Database System for Hotel Management System

INFO 5707 – DATA MODELLING FOR INFORMATIONAL PROFESSIONALS

Group 7
Professor: Abdullah Alenezi
Department of Information Sciences
University of North Texas

Team Members:

1. Tirumalesh Reddy, Neelapu - 11514998
2. Abhishek Puppala - 11596928
3. Greeshmanth Reddy Malapati - 11596210
4. Jashwanth Kalyan Polavarapu - 11596929
5. Manisha Lokasani - 11609309

Objective:

The objective of the project is to provide an overview of the data modelling process for a Hotel Management System. It will describe the purpose of data modelling, the key stakeholders involved, and the various steps involved in the data modelling process. Additionally, the document will discuss the various data models that can be used for a Hotel Management System, such as the Entity-Relationship (ER) model and provide guidance on how to create an effective data model for the system. Finally, the document will provide some examples of queries and reports that can be generated from the data model to support business decision making. The aim of this document is to assist informational professionals, such as data analysts, data scientists, and database administrators, in understanding the principles of data modelling for Hotel Management System and to help them create effective data models to support their business needs.

Scope:

We are building a hotel management System, which can be utilized by any Hotel to manage their data instead of working with audit books or handwritten transactions. The database system is primarily focusing on help tracking the Rooms Inventory, Reservations

Requirements:

In the database that we built, we can store and retrieve different attributes or data related to managing a hotel. It is a simple and easy-to-understand database and data model can be integrated with other environments like a web application or an API to serve the data as requested by the end user. The data model is built without any data redundancy and following the Normalization concepts in mind.

We have worked on the SQL Server Developer edition as our Relational Database Management System (RDBMS) for creating the database and SQL Server Management Studio (SSMS) as the IDE or client to interact with the database like query executions and viewing the metadata of the database.

Data Services & Tools:

- SQL Server Developer Edition (RDBMS)
- SQL Server Management Studio (SSMS – IDE)

Business Rules:

Following are the business rules that helped us with the design and development of the Data Model, ensuring that it meets the requirements of the hotel chain and its customers.

- Customers can book multiple rooms for different dates, but they cannot book the same room for overlapping dates.
- Rooms must be clean and available for occupancy before check-in time.
- Staff must clean rooms and update the availability status before check-in time.
- Customers must pay the full reservation amount before checking in.
- Staff must enter the correct room rate and room capacity information for each room.
- Room rates may vary by room type and season.
- Customers can cancel a reservation up to 24 hours before the check-in date.
- Staff must update the reservation status and refund the customer's payment if a reservation is cancelled.
- Rewards program members can earn points for each reservation and redeem points for discounts for free stays.
- Staff must update the rewards points and expiration date for each rewards program member.
- Payment methods include credit card, cash, and reward points redemption.
- Housekeeping staff must clean each room at least once a day and update the cleaning date.
- Event bookings can be made for different types of events, such as conferences, weddings, and parties.
- Event rooms must be available and booked in advance, and the total cost of the event must be paid before the vent date.
- Staff must update the event booking status and issue refunds if an event is cancelled.
- The system must be able to generate reports on room occupancy rates, revenue and other performance metrics for the hotel.

Entity Relationship (ER) Diagram:

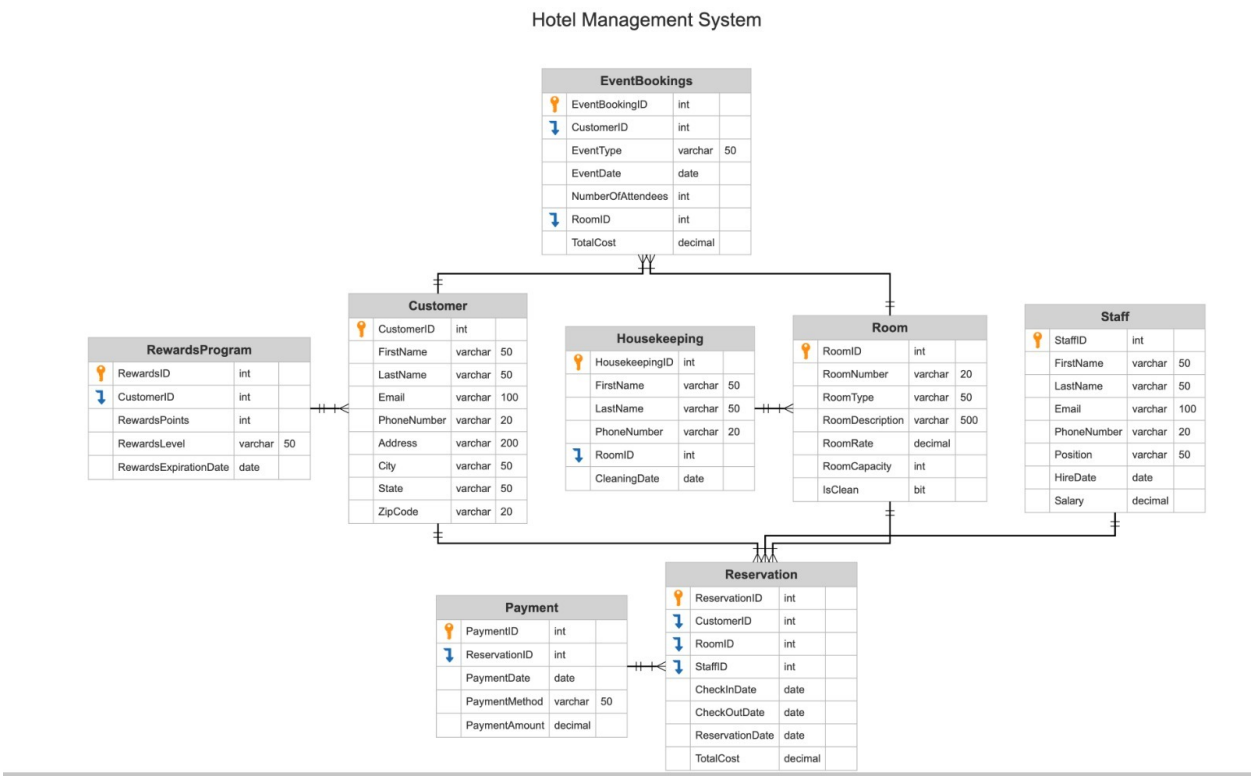

Figure 1: ER Diagram for Hotel Management System

Data Dictionary:

- Customer Table
  - CustomerID: The unique identifier for the customer.
  - FirstName: The first name of the customer.
  - LastName: The last name of the customer.
  - Email: The email address of the customer.
  - PhoneNumber: The phone number of the customer.
  - Address: The address of the customer.
  - City: The city where the customer resides.
  - State: The state where the customer resides.
  - ZipCode: The zip code where the customer resides.

- Staff Table
  - StaffID: The unique identifier for the staff member.
  - FirstName: The first name of the staff member.
  - LastName: The last name of the staff member.
  - Email: The email address of the staff member.
  - PhoneNumber: The phone number of the staff member.
  - Position: The position held by the staff member.
  - HireDate: The date the staff member was hired.

    - Salary: The salary of the staff member.

- Room Table
    - RoomID: The unique identifier for the room.
    - RoomNumber: The room number.
    - RoomType: The type of room (e.g. single, double, suite).
    - RoomDescription: A description of the room.
    - RoomRate: The rate charged for the room.
    - RoomCapacity: The maximum number of guests that can occupy the room.
    - IsClean: A flag indicating whether the room is clean or not.

- Reservation Table
    - ReservationID: The unique identifier for the reservation.
    - CustomerID: The unique identifier for the customer who made the reservation.
    - RoomID: The unique identifier for the room that was reserved.
    - StaffID: The unique identifier for the staff member who assisted with the reservation.
    - CheckInDate: The date the reservation begins.
    - CheckOutDate: The date the reservation ends.
    - ReservationDate: The date the reservation was made.
    - TotalCost: The total cost of the reservation.

- Payment Table
    - PaymentID: The unique identifier for the payment.
    - ReservationID: The unique identifier for the reservation that the payment is associated with.
    - PaymentDate: The date the payment was made.
    - PaymentMethod: The payment method used (e.g. credit card, cash).
    - PaymentAmount: The amount paid.

- Housekeeping Table
    - HousekeepingID: The unique identifier for the housekeeper.
    - FirstName: The first name of the housekeeper.
    - LastName: The last name of the housekeeper.
    - PhoneNumber: The phone number of the housekeeper.
    - RoomID: The unique identifier for the room that was cleaned.
    - CleaningDate: The date the room was cleaned.

- Rewards Program Table
    - RewardsID: The unique identifier for the rewards program entry.
    - CustomerID: The unique identifier for the customer who is enrolled in the rewards program.
    - RewardsPoints: The number of rewards points the customer has earned.
    - RewardsLevel: The level of the customer in the rewards program.

- RewardsExpirationDate: The date when the customer's rewards will expire.

- Events Booking Table
  - EventBookingID: The unique identifier for the event booking.
  - CustomerID: The unique identifier for the customer who made the event booking.
  - EventType: The type of event (e.g. wedding, business meeting).
  - EventDate: The date of the event.
  - NumberOfAttendees: The number of attendees expected at the event.
  - RoomID: The unique identifier for the room
  - TotalCost: The total cost of the event booking.

Queries & Operations:

a. Customer Table

1. Create Table Statement

```
CREATE TABLE Customer (
        CustomerID INT PRIMARY KEY,
        FirstName VARCHAR(50),
        LastName VARCHAR(50),
        Email VARCHAR(100),
        PhoneNumber VARCHAR(20),
        Address VARCHAR(200),
        City VARCHAR(50),
        State VARCHAR(50),
        ZipCode VARCHAR(20)
);
```

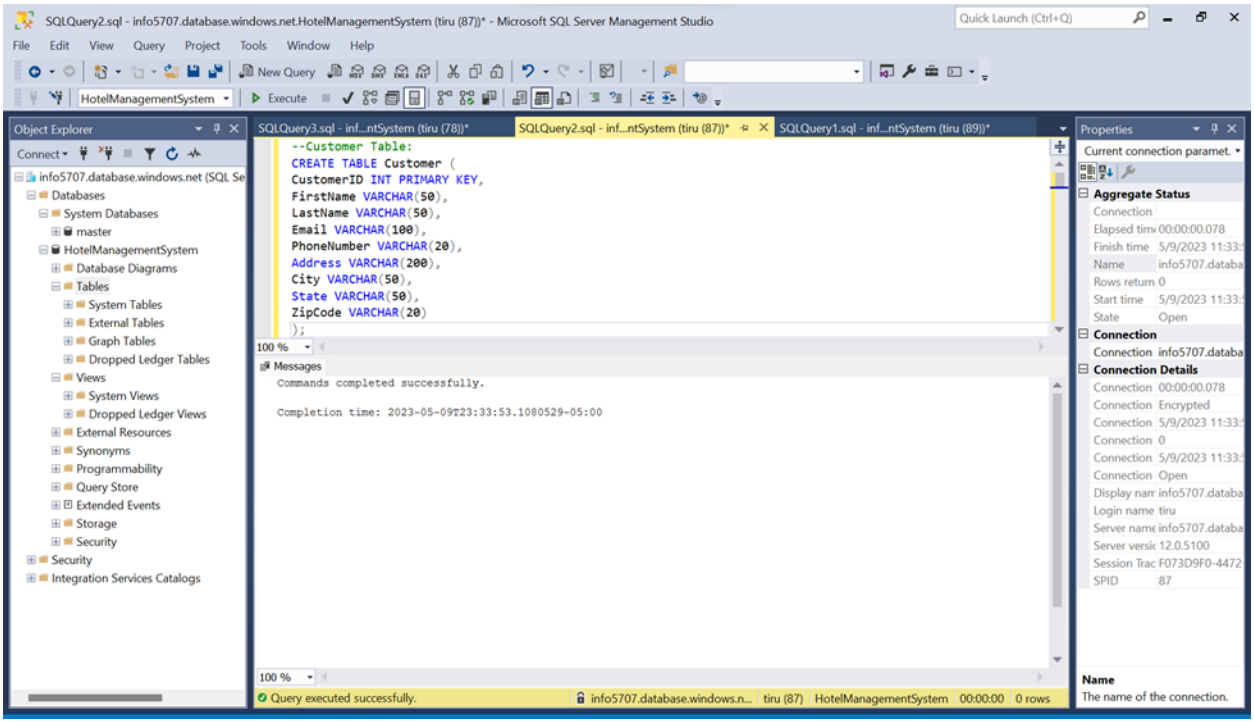

Figure 2: Cutomer Table Creation

2. Insert Scripts

INSERT INTO Customer (CustomerID, FirstName, LastName, Email, PhoneNumber, Address, City, State, ZipCode) VALUES
(1, 'John', 'Doe', 'johndoe@example.com', '555-1234', '123 Main St', 'Anytown', 'CA', '12345'),
(2, 'Jane', 'Smith', 'janesmith@example.com', '555-5678', '456 Elm St', 'Anycity', 'TX', '67890'),
(3, 'Bob', 'Johnson', 'bobjohnson@example.com', '555-9012', '789 Oak St', 'Anywhere', 'FL', '34567'),
(4, 'Mary', 'Jones', 'maryjones@example.com', '555-3456', '234 Maple St', 'Anyville', 'NY', '89012'),
(5, 'Mike', 'Williams', 'mikewilliams@example.com', '555-7890', '567 Pine St', 'Anystate', 'WA', '45678'),
(6, 'Karen', 'Brown', 'karenbrown@example.com', '555-2345', '890 Cedar St', 'Anycity', 'IL', '12345'),
(7, 'Tom', 'Davis', 'tomdavis@example.com', '555-6789', '123 Walnut St', 'Anytown', 'MI', '67890'),
(8, 'Susan', 'Garcia', 'susangarcia@example.com', '555-0123', '456 Oak St', 'Anycity', 'OH', '34567'),
(9, 'David', 'Lee', 'davidlee@example.com', '555-4567', '789 Maple St', 'Anywhere', 'PA', '89012'),
(10, 'Lisa', 'Martinez', 'lisamartinez@example.com', '555-8901', '234 Cedar St', 'Anyville', 'GA', '45678'),

(11, 'Mark', 'Taylor', 'marktaylor@example.com', '555-2345', '567 Pine St', 'Anystate', 'NC', '12345'),
(12, 'Julie', 'Hernandez', 'juliehernandez@example.com', '555-6789', '890 Walnut St', 'Anytown', 'VA', '67890'),
(13, 'Kevin', 'Young', 'kevinyoung@example.com', '555-0123', '123 Maple St', 'Anycity', 'NJ', '34567'),
(14, 'Emily', 'Allen', 'emilyallen@example.com', '555-4567', '456 Cedar St', 'Anywhere', 'MA', '89012'),
(15, 'Adam', 'Brown', 'adambrown@example.com', '555-8901', '789 Oak St', 'Anycity', 'KY', '45678');

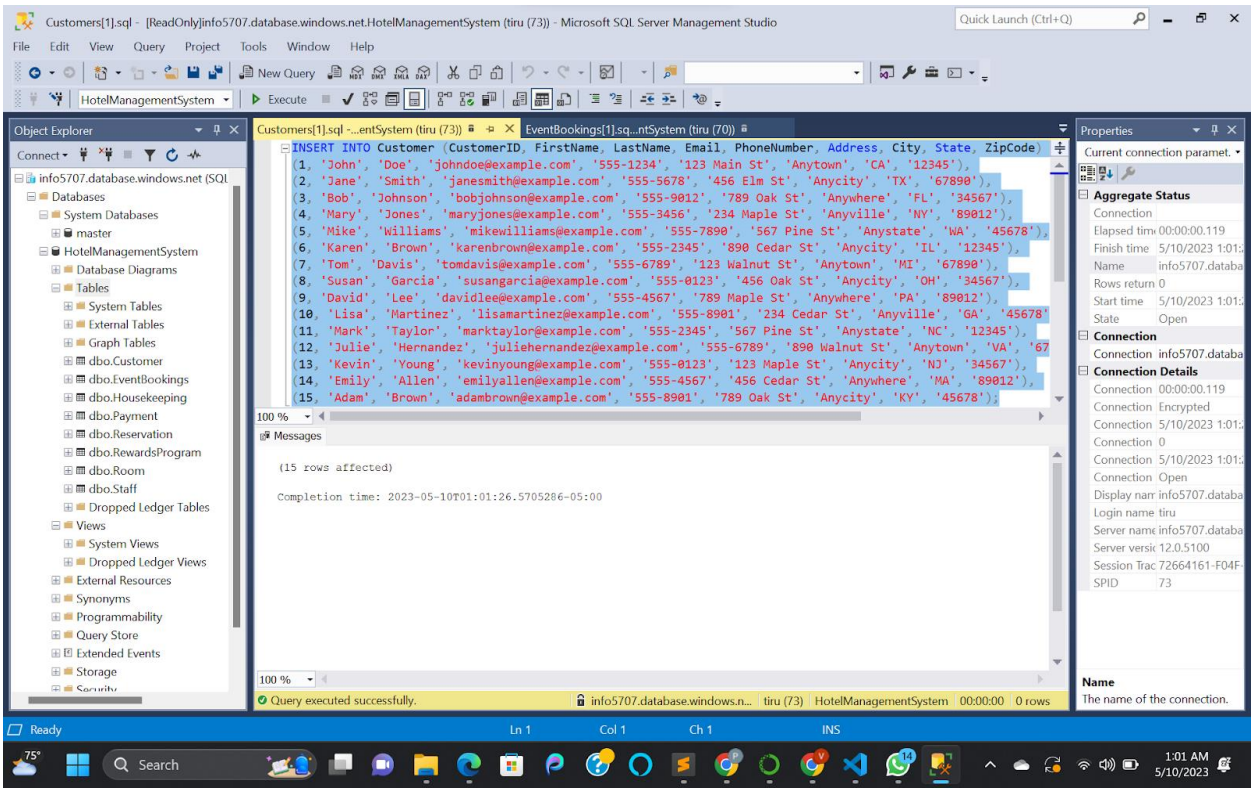

Figure 3: Customer Insertion Scripts

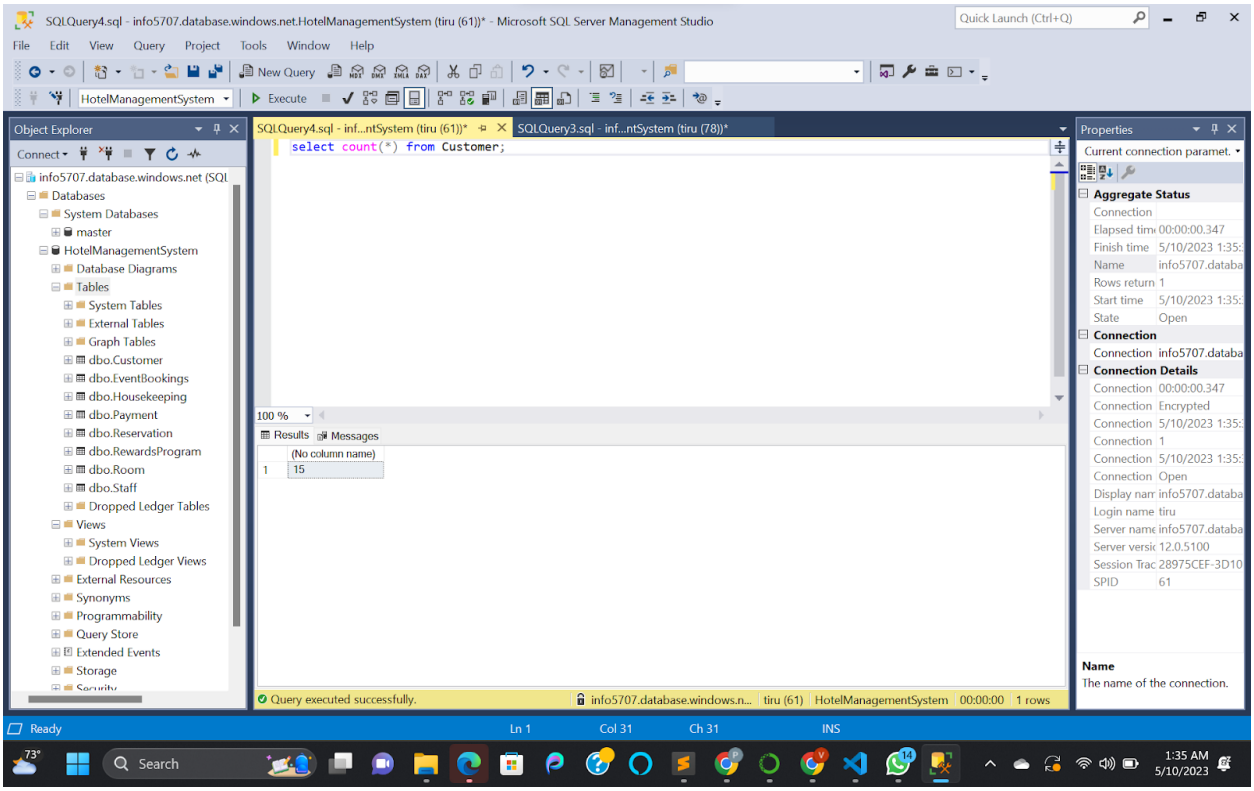

Figure 4: Customer Table Row Count

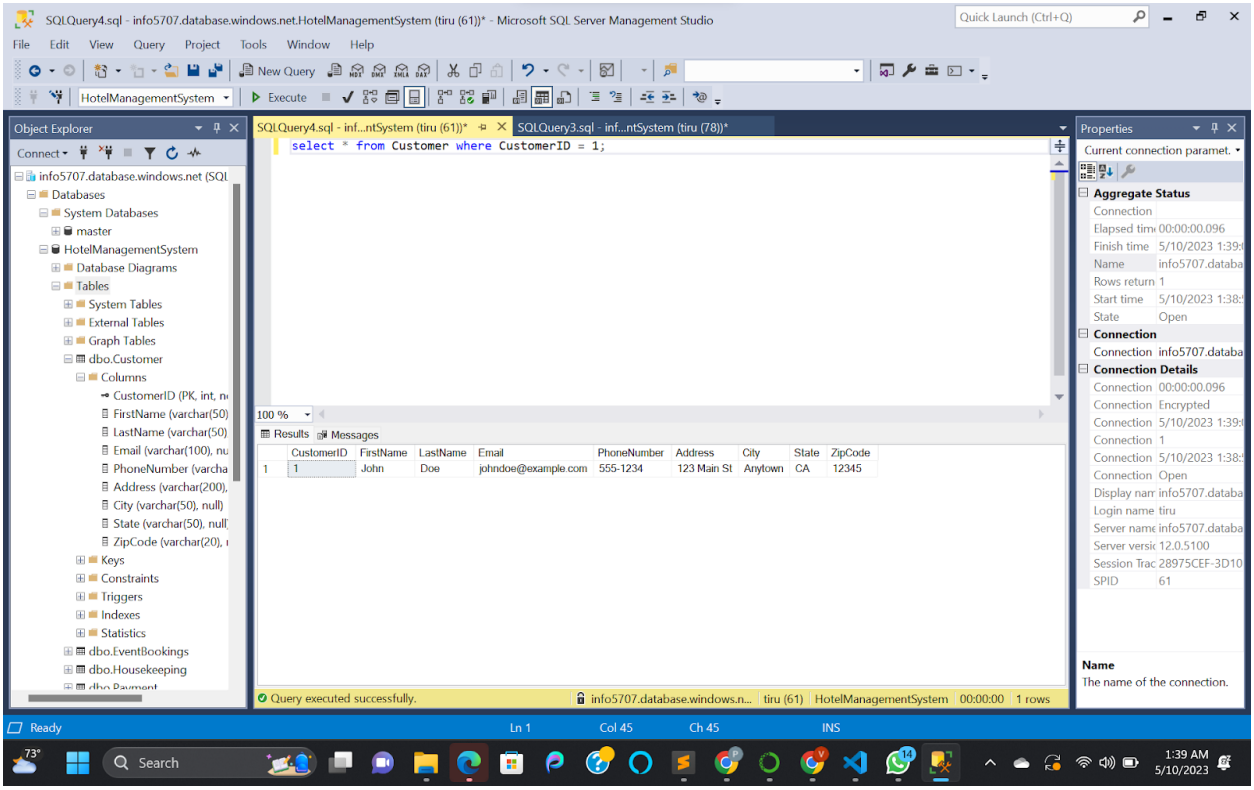

Figure 5: Customer Table Filter Query

3. Update Statement:

```
UPDATE Customer
SET FirstName = 'Alfred', LastName = 'Frank'
WHERE CustomerID = 1;
```

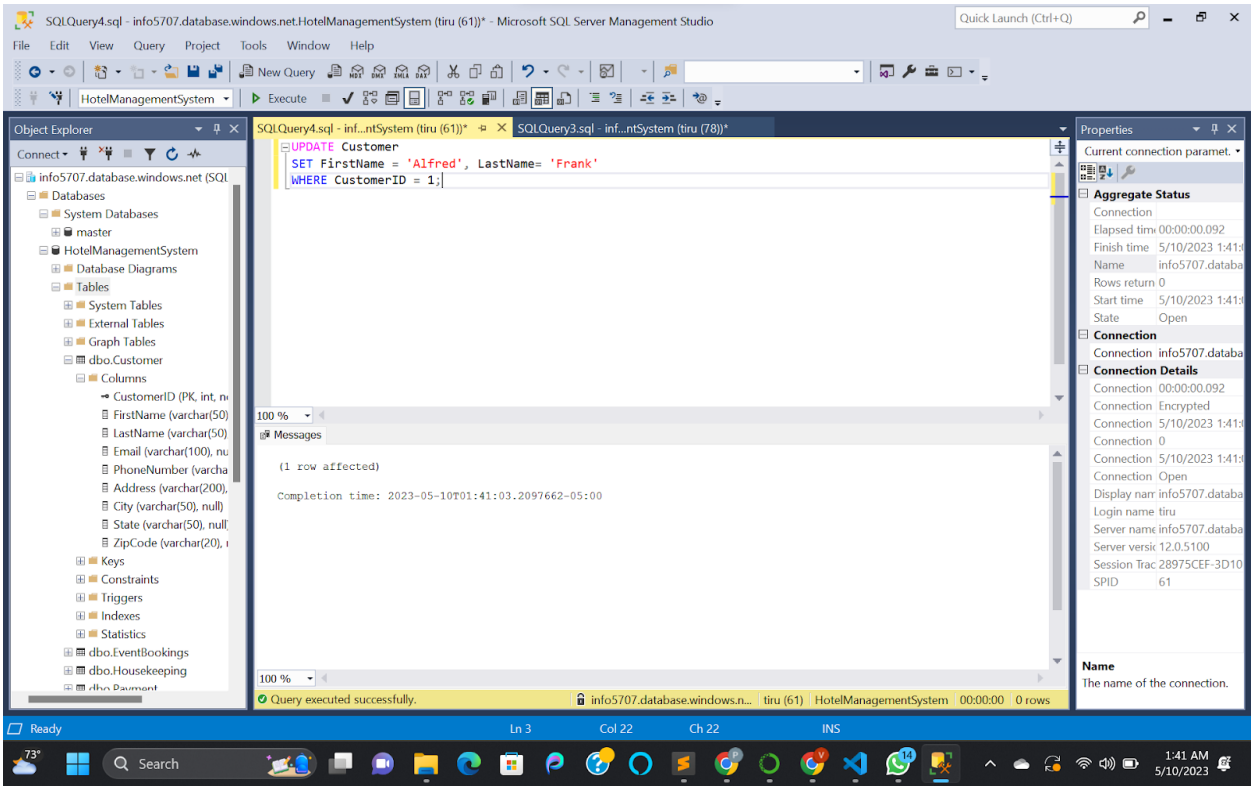

Figure 6: Update Statement for Customer Table

b. Staff Table

1. Create Table Statement

```
CREATE TABLE Staff (
        StaffID INT PRIMARY KEY,
        FirstName VARCHAR(50),
        LastName VARCHAR(50),
        Email VARCHAR(100),
        PhoneNumber VARCHAR(20),
        Position VARCHAR(50),
        HireDate DATE,
        Salary DECIMAL(10,2)
);
```

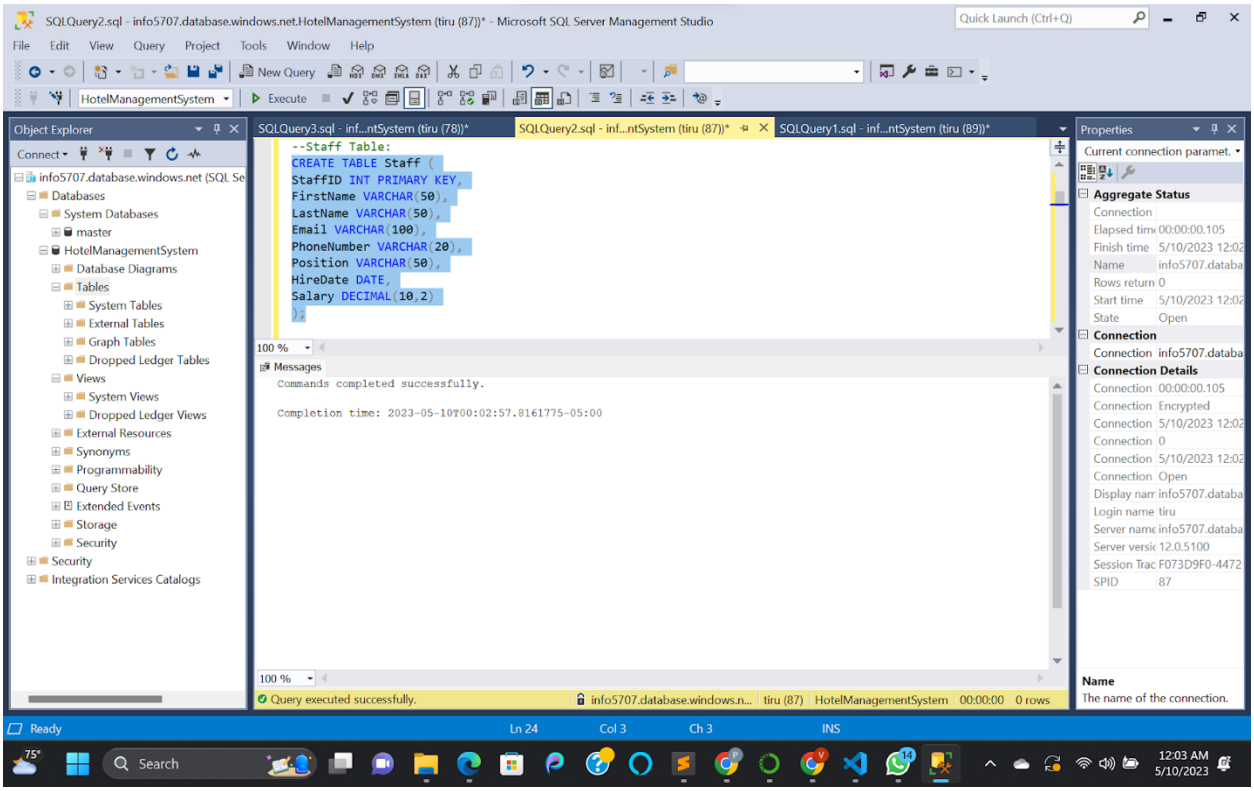

Figure 7: Staff Table Creation

2. Insert Scripts

INSERT INTO Staff (StaffID, FirstName, LastName, Email, PhoneNumber, Position, HireDate, Salary) VALUES
(1, 'John', 'Smith', 'johnsmith@example.com', '555-1234', 'Manager', '2020-01-01', 60000.00),
(2, 'Jane', 'Doe', 'janedoe@example.com', '555-5678', 'Receptionist', '2020-02-01', 35000.00),
(3, 'Bob', 'Johnson', 'bobjohnson@example.com', '555-9012', 'Staff', '2020-03-01', 40000.00),
(4, 'Mary', 'Williams', 'marywilliams@example.com', '555-3456', 'Housekeeping', '2020-04-01', 30000.00),
(5, 'Mike', 'Jones', 'mikejones@example.com', '555-7890', 'Maintenance', '2020-05-01', 35000.00),
(6, 'Karen', 'Garcia', 'karengarcia@example.com', '555-2345', 'Chef', '2020-06-01', 50000.00),
(7, 'Tom', 'Lee', 'tomlee@example.com', '555-6789', 'Waitstaff', '2020-07-01', 25000.00),
(8, 'Susan', 'Martinez', 'susanmartinez@example.com', '555-0123', 'Bartender', '2020-08-01', 30000.00),
(9, 'David', 'Taylor', 'davidtaylor@example.com', '555-4567', 'Housekeeping', '2020-09-01', 30000.00),

(10, 'Lisa', 'Hernandez', 'lisahernandez@example.com', '555-8901', 'Waitstaff', '2020-10-01', 25000.00),
(11, 'Mark', 'Brown', 'markbrown@example.com', '555-2345', 'Maintenance', '2020-11-01', 35000.00),
(12, 'Julie', 'Young', 'julieyoung@example.com', '555-6789', 'Receptionist', '2020-12-01', 35000.00),
(13, 'Kevin', 'Allen', 'kevinallen@example.com', '555-0123', 'Manager', '2021-01-01', 70000.00),
(14, 'Emily', 'Davis', 'emilydavis@example.com', '555-4567', 'Chef', '2021-02-01', 50000.00),
(15, 'Adam', 'Garcia', 'adamgarcia@example.com', '555-8901', 'Bartender', '2021-03-01', 30000.00);

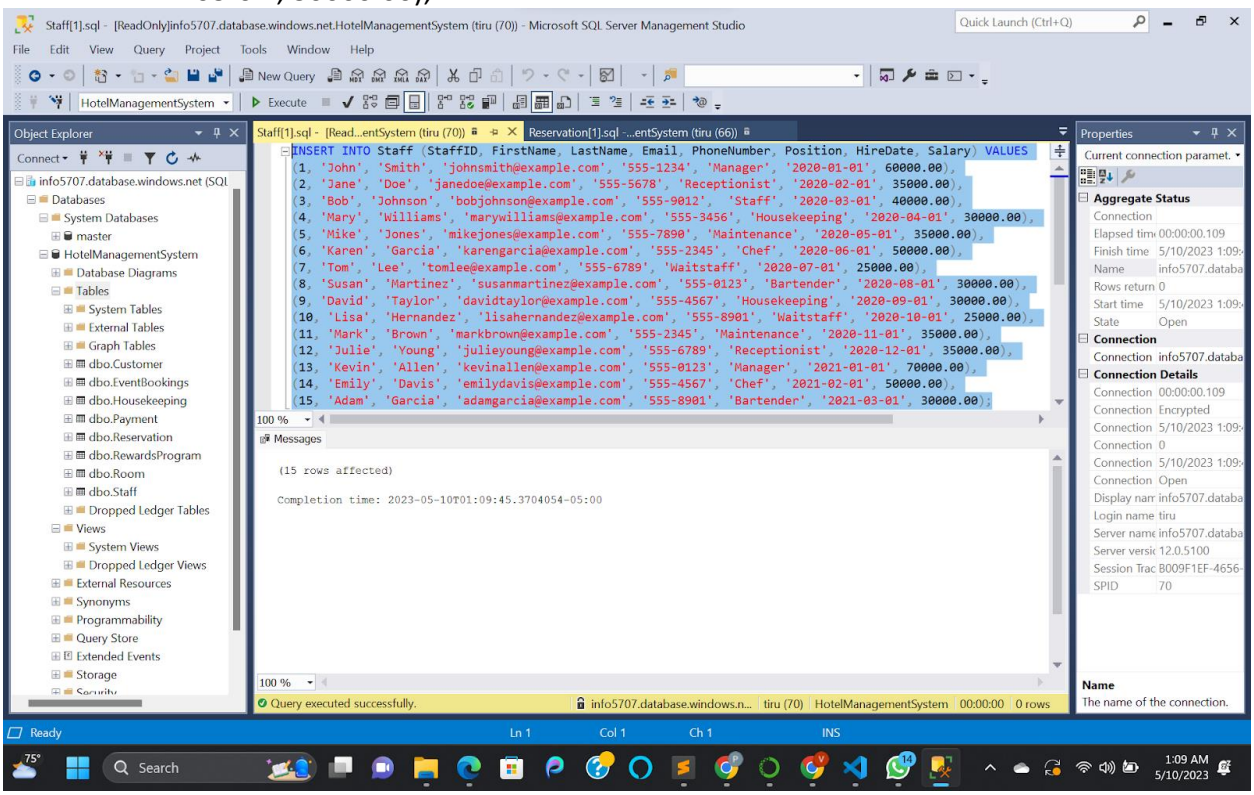

Figure 8: Insert Statement for Staff Table

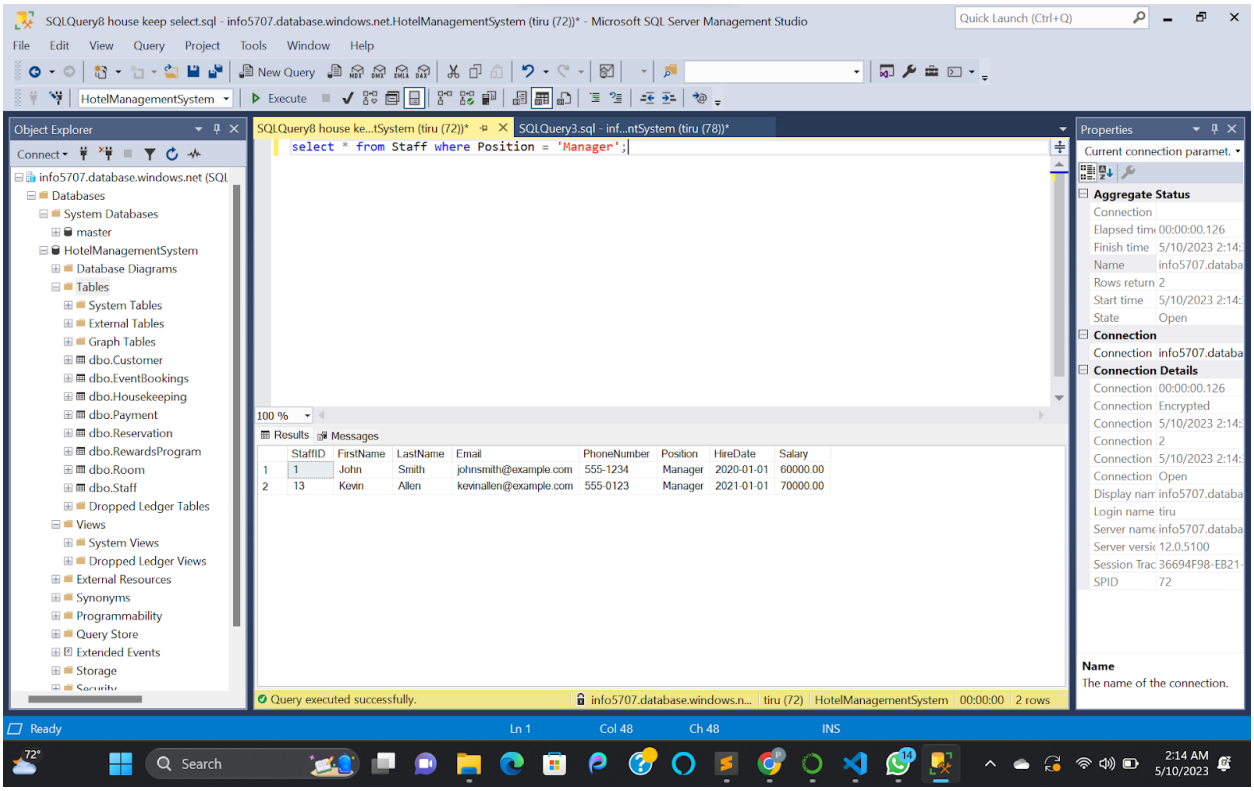

Figure 9: Select Statement for Staff Table

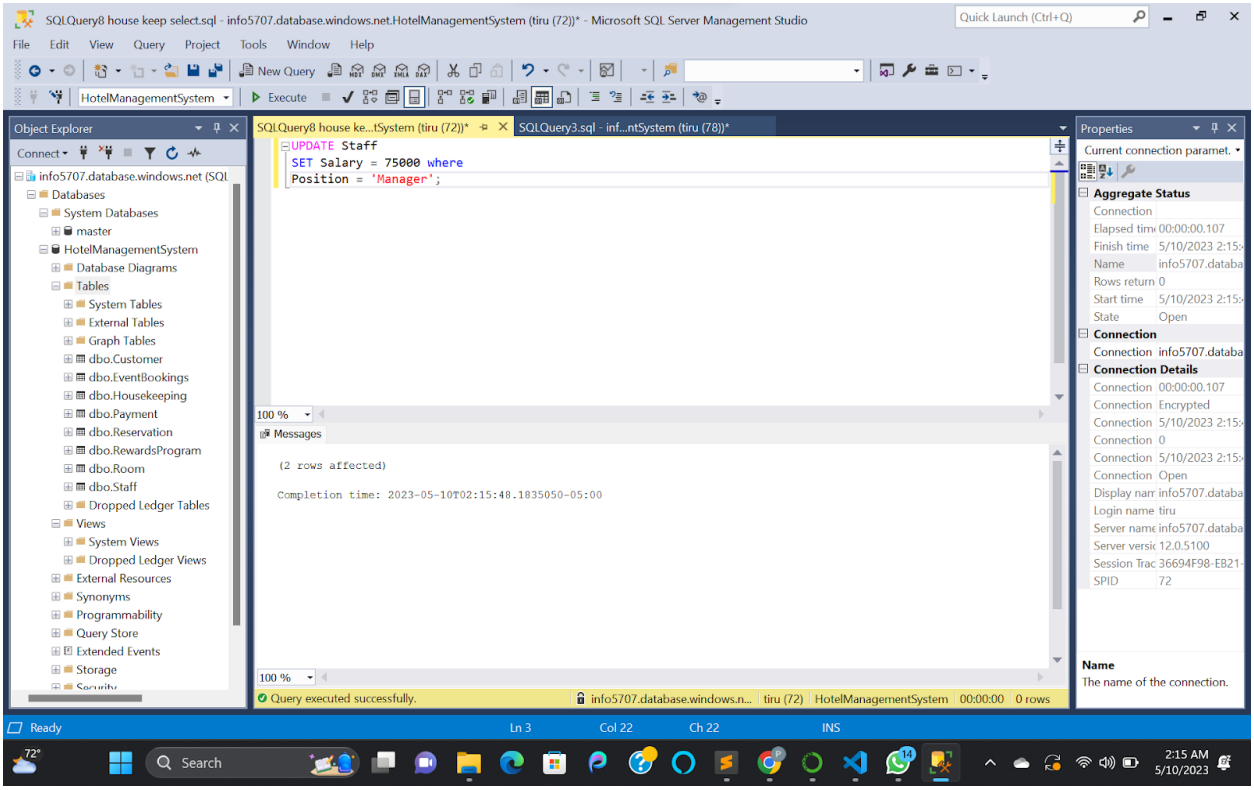

Figure 10: Updating Manager Salary

c. Room Table

1. Create Table Statement

```
CREATE TABLE Room (
    RoomID INT PRIMARY KEY,
    RoomNumber VARCHAR(20),
    RoomType VARCHAR(50),
    RoomDescription VARCHAR(500),
    RoomRate DECIMAL(10,2),
    RoomCapacity INT,
    IsClean BIT
);
```

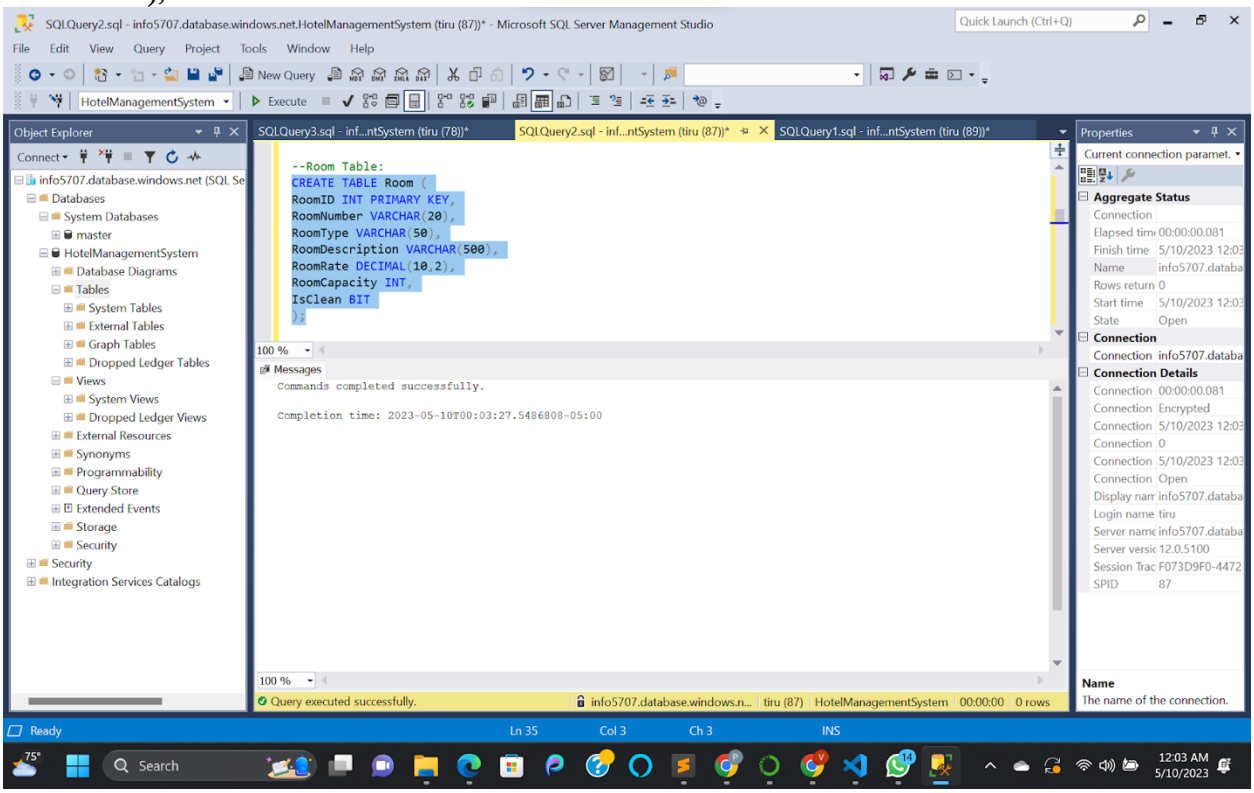

Figure 11: Room Table Creation

2. Insert Scripts

```
INSERT INTO Room (RoomID, RoomNumber, RoomType, RoomDescription,
RoomRate, RoomCapacity, IsClean) VALUES
(1, '101', 'Standard', 'Standard Room with 1 Queen Bed', 100.00, 2, 1),
(2, '102', 'Standard', 'Standard Room with 2 Queen Beds', 120.00, 4, 0),
(3, '103', 'Deluxe', 'Deluxe Room with 1 King Bed', 150.00, 2, 1),
(4, '104', 'Deluxe', 'Deluxe Room with 2 Double Beds', 160.00, 4, 1),
(5, '105', 'Suite', 'Suite with 1 King Bed and Jacuzzi', 200.00, 2, 0),
(6, '106', 'Suite', 'Suite with 2 Queen Beds and Kitchen', 220.00, 4, 1),
(7, '107', 'Standard', 'Standard Room with 1 Queen Bed', 100.00, 2, 1),
```

(8, '108', 'Standard', 'Standard Room with 2 Double Beds', 120.00, 4, 0),
(9, '109', 'Deluxe', 'Deluxe Room with 1 King Bed', 150.00, 2, 1),
(10, '110', 'Deluxe', 'Deluxe Room with 2 Queen Beds', 160.00, 4, 1),
(11, '111', 'Suite', 'Suite with 1 King Bed and Jacuzzi', 200.00, 2, 0),
(12, '112', 'Suite', 'Suite with 2 Queen Beds and Kitchen', 220.00, 4, 1),
(13, '113', 'Standard', 'Standard Room with 1 Queen Bed', 100.00, 2, 1),
(14, '114', 'Standard', 'Standard Room with 2 Queen Beds', 120.00, 4, 0),
(15, '115', 'Deluxe', 'Deluxe Room with 1 King Bed', 150.00, 2, 1),
(16, '116', 'Deluxe', 'Deluxe Room with 2 Double Beds', 160.00, 4, 1),
(17, '117', 'Suite', 'Suite with 1 King Bed and Jacuzzi', 200.00, 2, 0),
(18, '118', 'Suite', 'Suite with 2 Queen Beds and Kitchen', 220.00, 4, 1),
(19, '119', 'Standard', 'Standard Room with 1 Queen Bed', 100.00, 2, 1),
(20, '120', 'Standard', 'Standard Room with 2 Double Beds', 120.00, 4, 0);

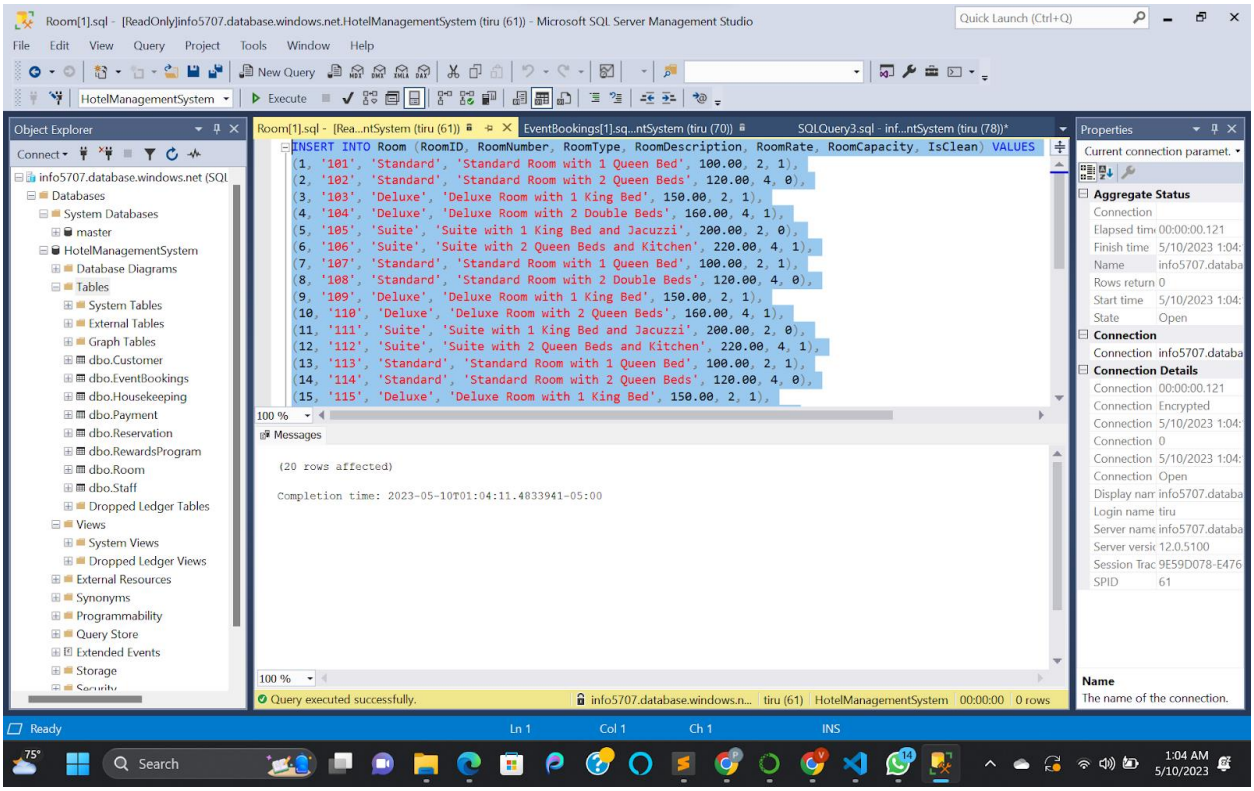

Figure 12: Room Table Insertions

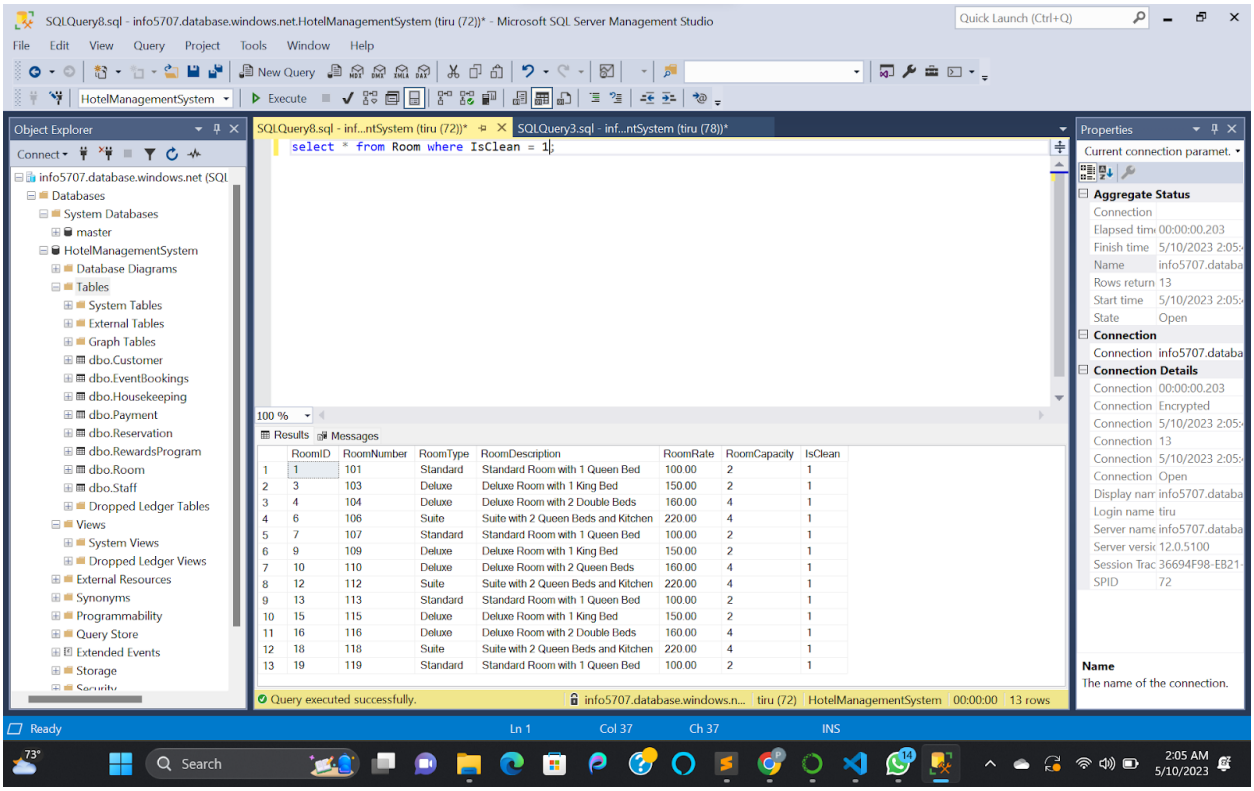

Figure 13: Select and Filter Condition on Room Table

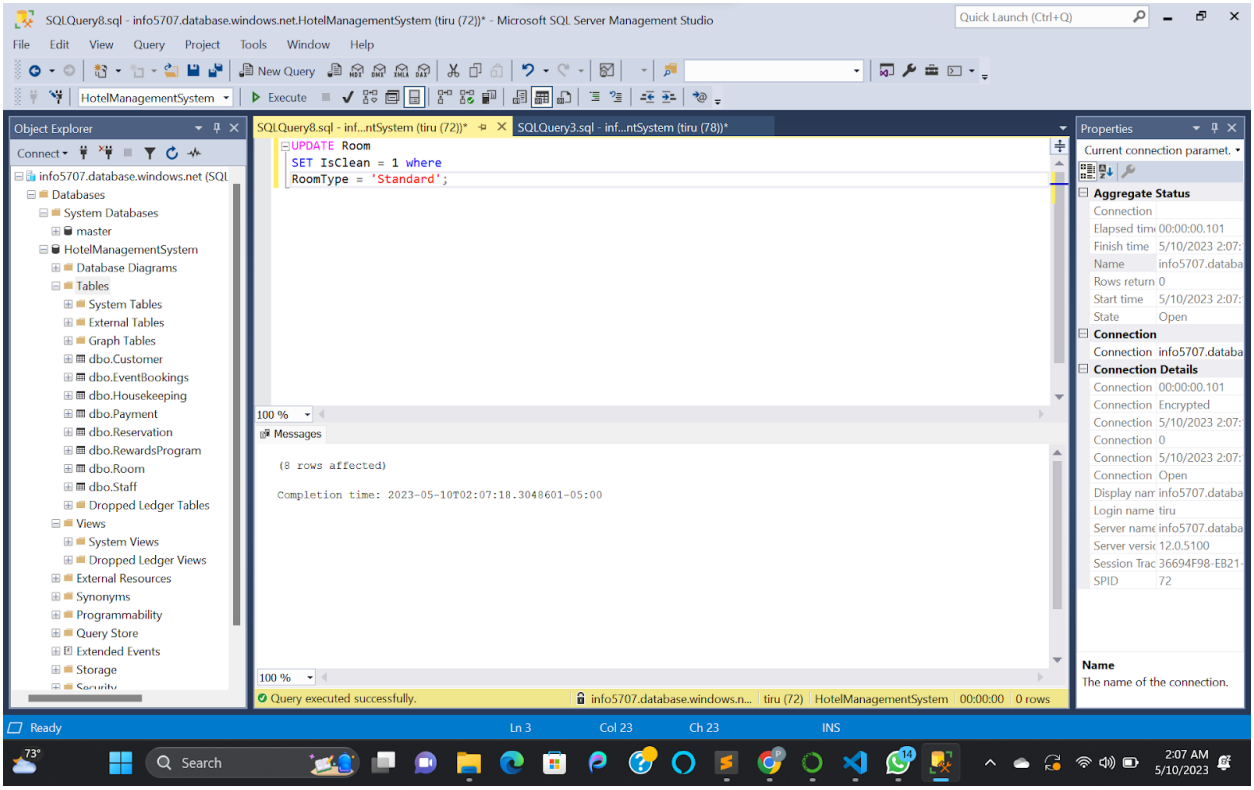

Figure 14: Update Statement on Room Table

d. Reservation Table

1. Create Table Statement

```
CREATE TABLE Reservation (
    ReservationID INT PRIMARY KEY,
    CustomerID INT,
    RoomID INT,
    StaffID INT,
    CheckInDate DATE,
    CheckOutDate DATE,
    ReservationDate DATE,
    TotalCost DECIMAL(10,2),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (RoomID) REFERENCES Room(RoomID),
    FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)
);
```

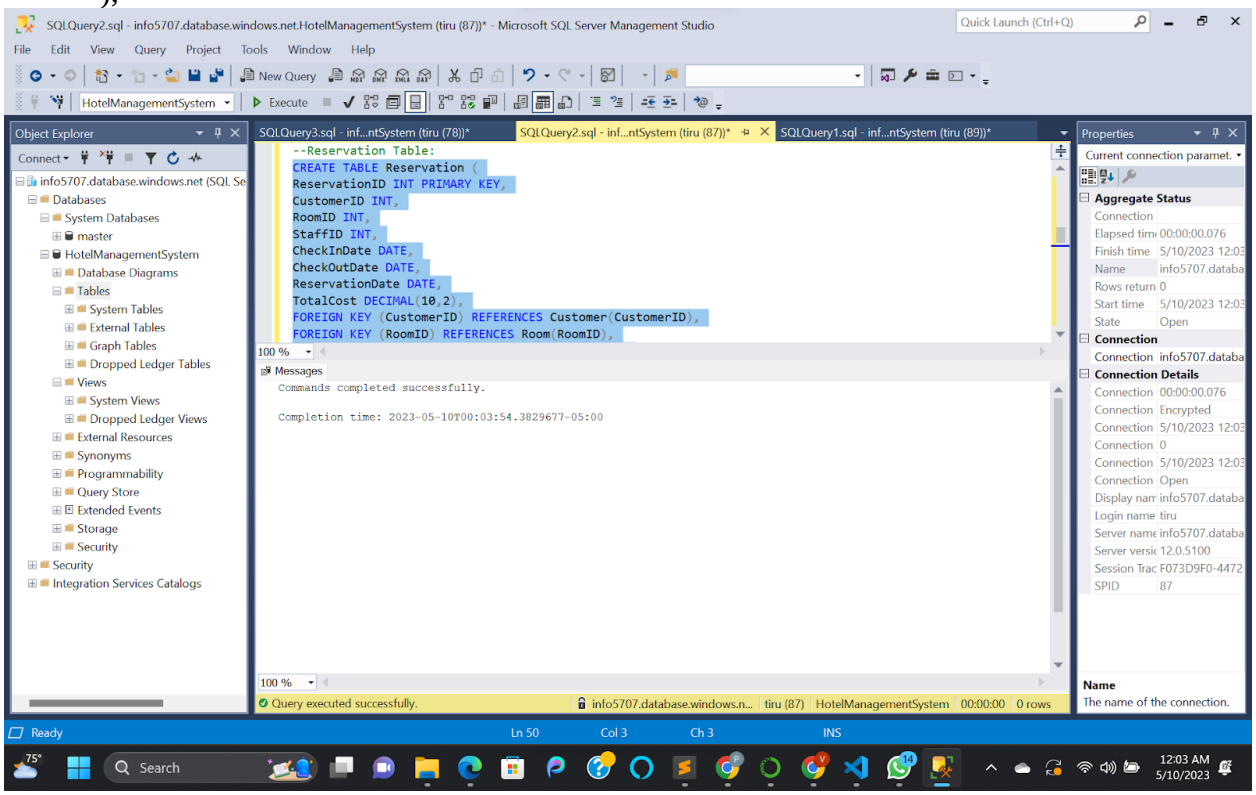

Figure 15: Reservation Table Creation

2. Insert Scripts

```
INSERT INTO Reservation (ReservationID, CustomerID, RoomID, StaffID, CheckInDate,
CheckOutDate, ReservationDate, TotalCost) VALUES
(1, 1, 1, 1, '2022-06-01', '2022-06-05', '2022-05-15', 400.00),
```

(2, 2, 2, 2, '2022-06-02', '2022-06-04', '2022-05-16', 240.00),
(3, 3, 3, 3, '2022-06-03', '2022-06-08', '2022-05-17', 750.00),
(4, 4, 4, 4, '2022-06-04', '2022-06-06', '2022-05-18', 320.00),
(5, 5, 5, 5, '2022-06-05', '2022-06-07', '2022-05-19', 400.00),
(6, 6, 6, 6, '2022-06-06', '2022-06-11', '2022-05-20', 1100.00),
(7, 7, 7, 7, '2022-06-07', '2022-06-08', '2022-05-21', 100.00),
(8, 8, 8, 8, '2022-06-08', '2022-06-10', '2022-05-22', 240.00),
(9, 9, 9, 9, '2022-06-09', '2022-06-14', '2022-05-23', 600.00),
(10, 10, 10, 10, '2022-06-10', '2022-06-12', '2022-05-24', 320.00),
(11, 11, 11, 11, '2022-06-11', '2022-06-14', '2022-05-25', 450.00),
(12, 12, 12, 12, '2022-06-12', '2022-06-13', '2022-05-26', 140.00),
(13, 13, 13, 13, '2022-06-13', '2022-06-17', '2022-05-27', 800.00),
(14, 14, 14, 14, '2022-06-14', '2022-06-16', '2022-05-28', 240.00),
(15, 15, 15, 15, '2022-06-15', '2022-06-20', '2022-05-29', 1100.00);

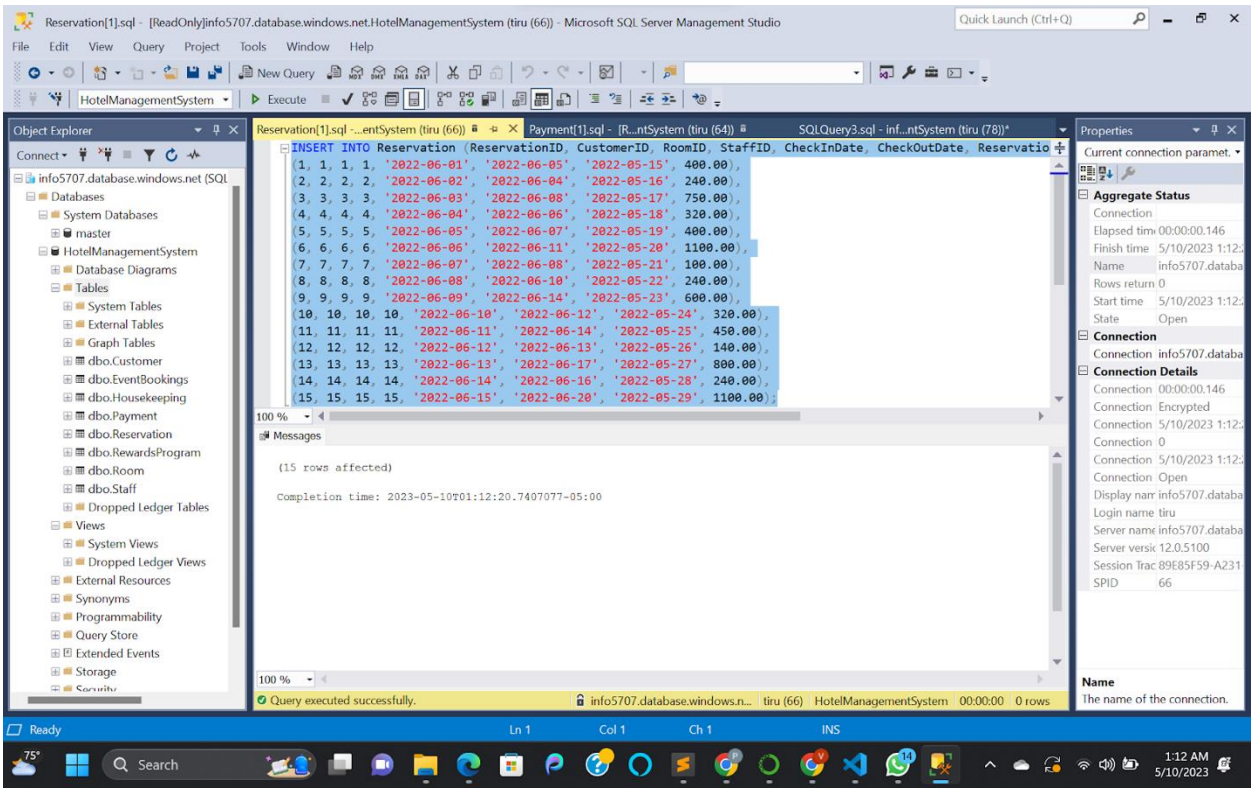

Figure 16: Reservation Table Insertions

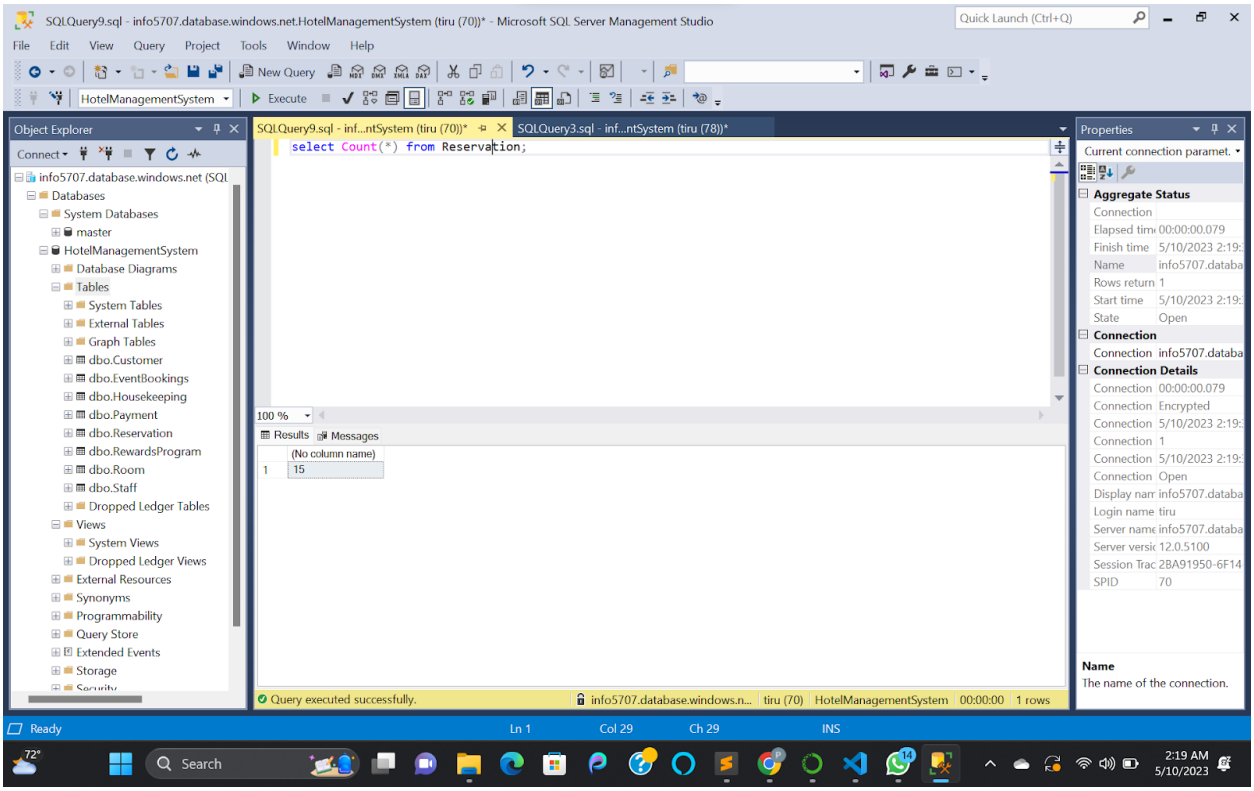

Figure 17: Reservation Table Row Count

e. Payment Table

1. Create Table Statement

```
CREATE TABLE Payment (
    PaymentID INT PRIMARY KEY,
    ReservationID INT,
    PaymentDate DATE,
    PaymentMethod VARCHAR(50),
    PaymentAmount DECIMAL(10,2),
    FOREIGN KEY (ReservationID) REFERENCES Reservation(ReservationID)
);
```

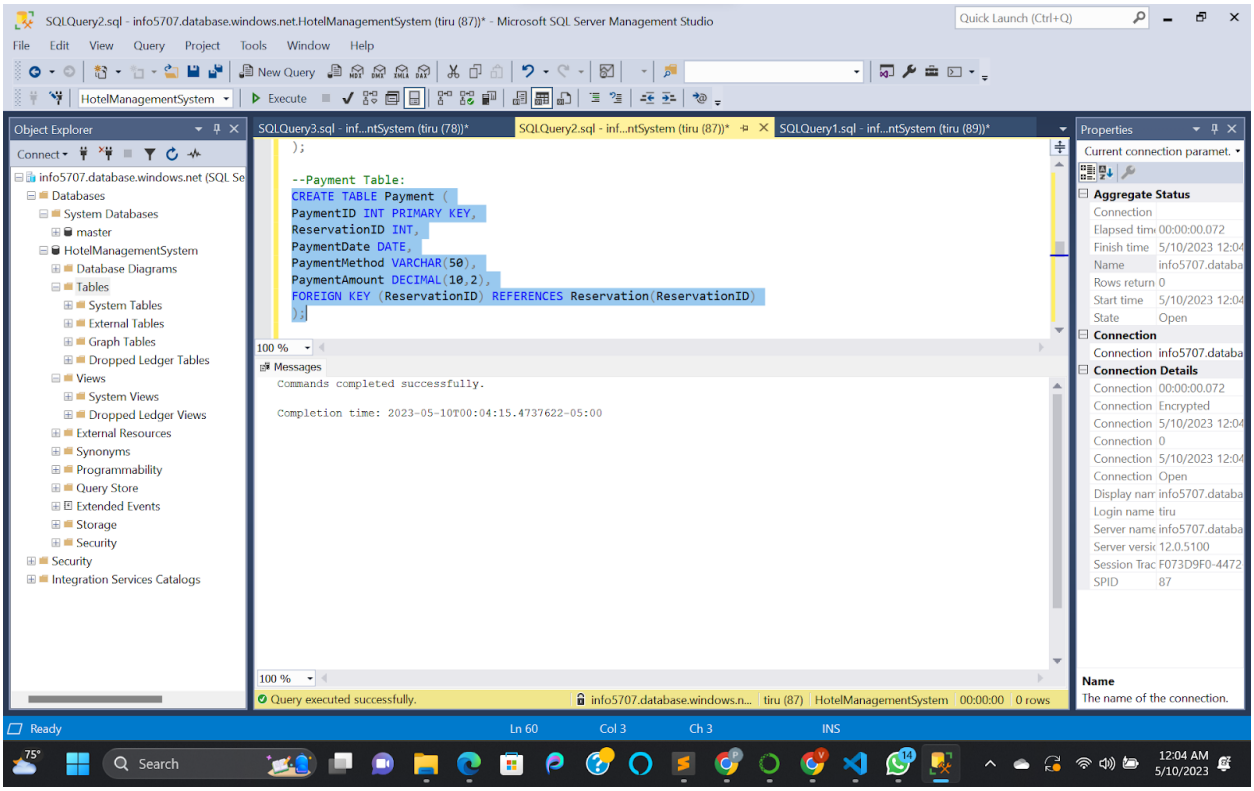

Figure 18: Payment Table Creation

2. Insert Scripts

INSERT INTO Payment (PaymentID, ReservationID, PaymentDate, PaymentMethod, PaymentAmount) VALUES
(1, 1, '2022-05-31', 'Credit Card', 400.00),
(2, 2, '2022-05-31', 'Cash', 240.00),
(3, 3, '2022-06-01', 'Credit Card', 750.00),
(4, 4, '2022-06-01', 'Cash', 320.00),
(5, 5, '2022-06-02', 'Credit Card', 400.00),
(6, 6, '2022-06-03', 'Credit Card', 1100.00),
(7, 7, '2022-06-04', 'Credit Card', 100.00),
(8, 8, '2022-06-05', 'Cash', 240.00),
(9, 9, '2022-06-05', 'Credit Card', 600.00),
(10, 10, '2022-06-06', 'Credit Card', 320.00),
(11, 11, '2022-06-06', 'Cash', 450.00),
(12, 12, '2022-06-07', 'Credit Card', 140.00),
(13, 13, '2022-06-08', 'Credit Card', 800.00),
(14, 14, '2022-06-09', 'Cash', 240.00),
(15, 15, '2022-06-10', 'Credit Card', 1100.00);

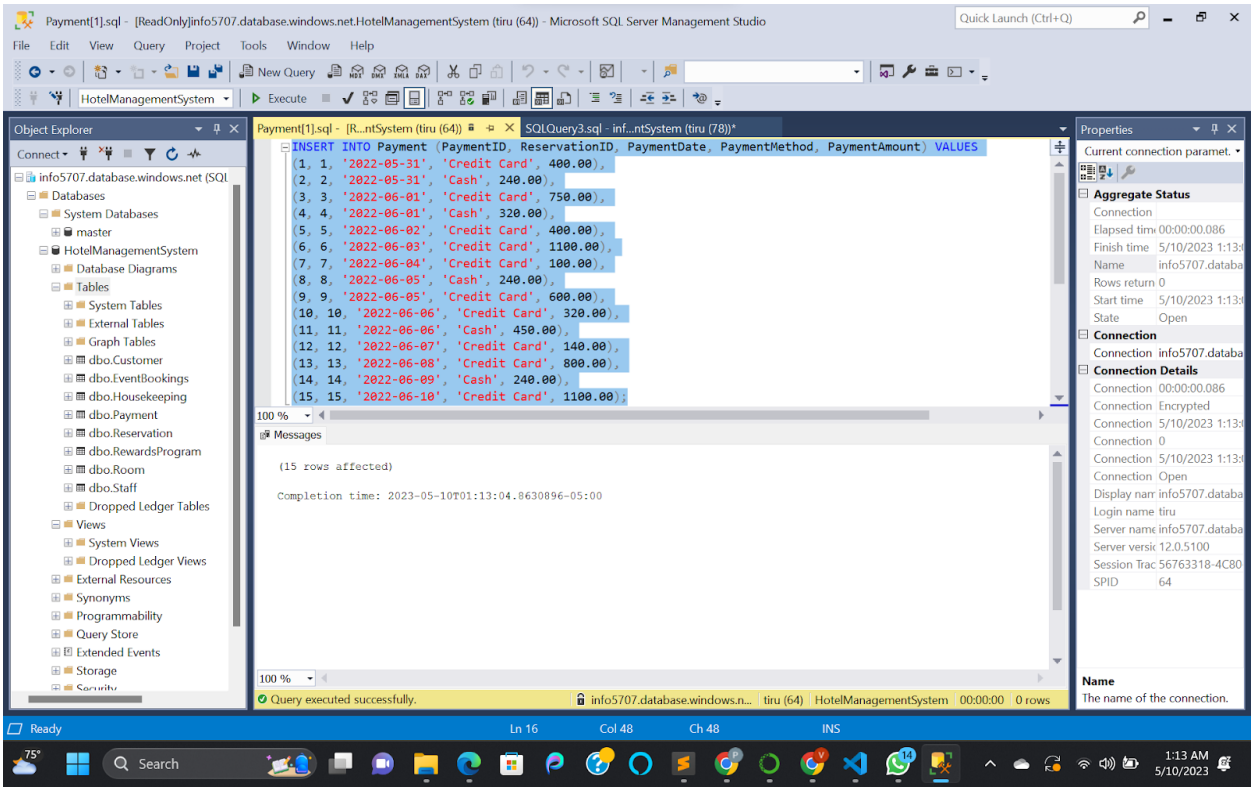

Figure 19: Payment Table Insertions

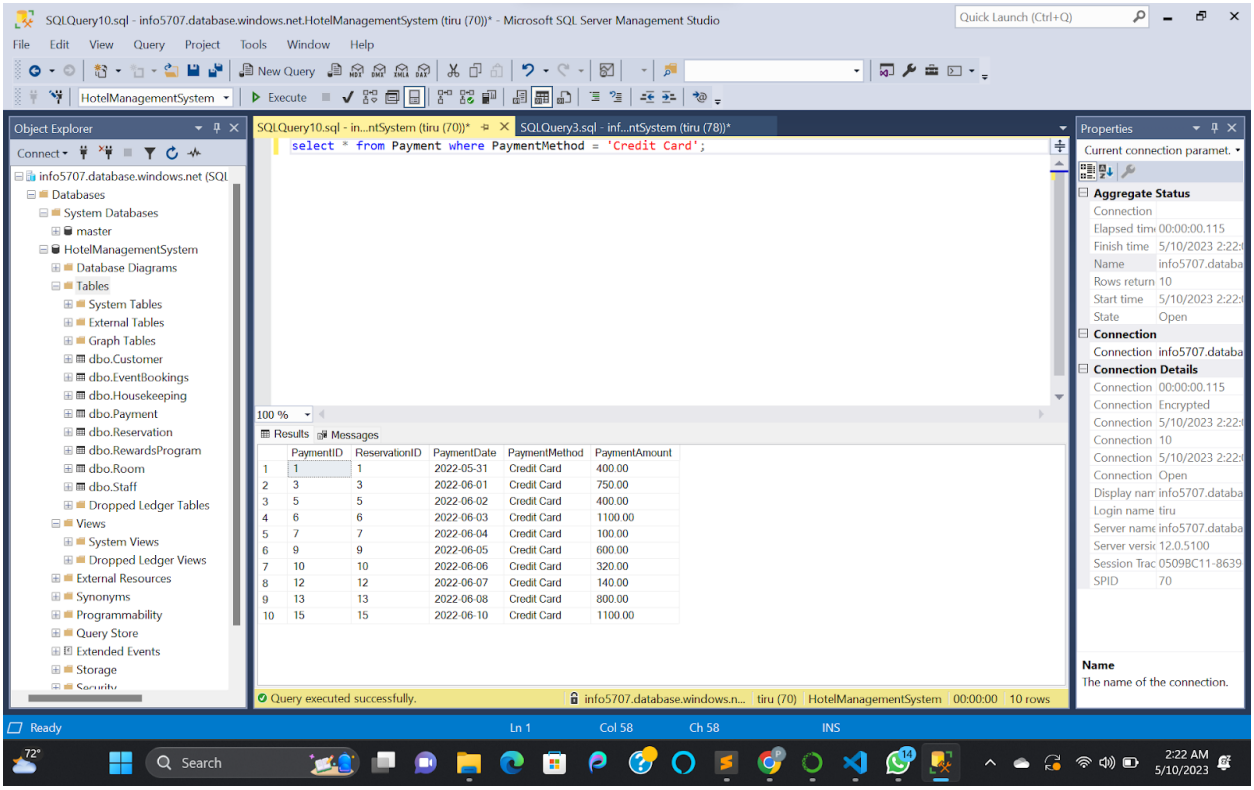

Figure 20: Payment Table Select and Filter Query

f. Housekeeping Table

1. Create Table Statement

```
CREATE TABLE Housekeeping (
    HousekeepingID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    PhoneNumber VARCHAR(20),
    RoomID INT,
    CleaningDate DATE,
    FOREIGN KEY (RoomID) REFERENCES Room(RoomID)
);
```

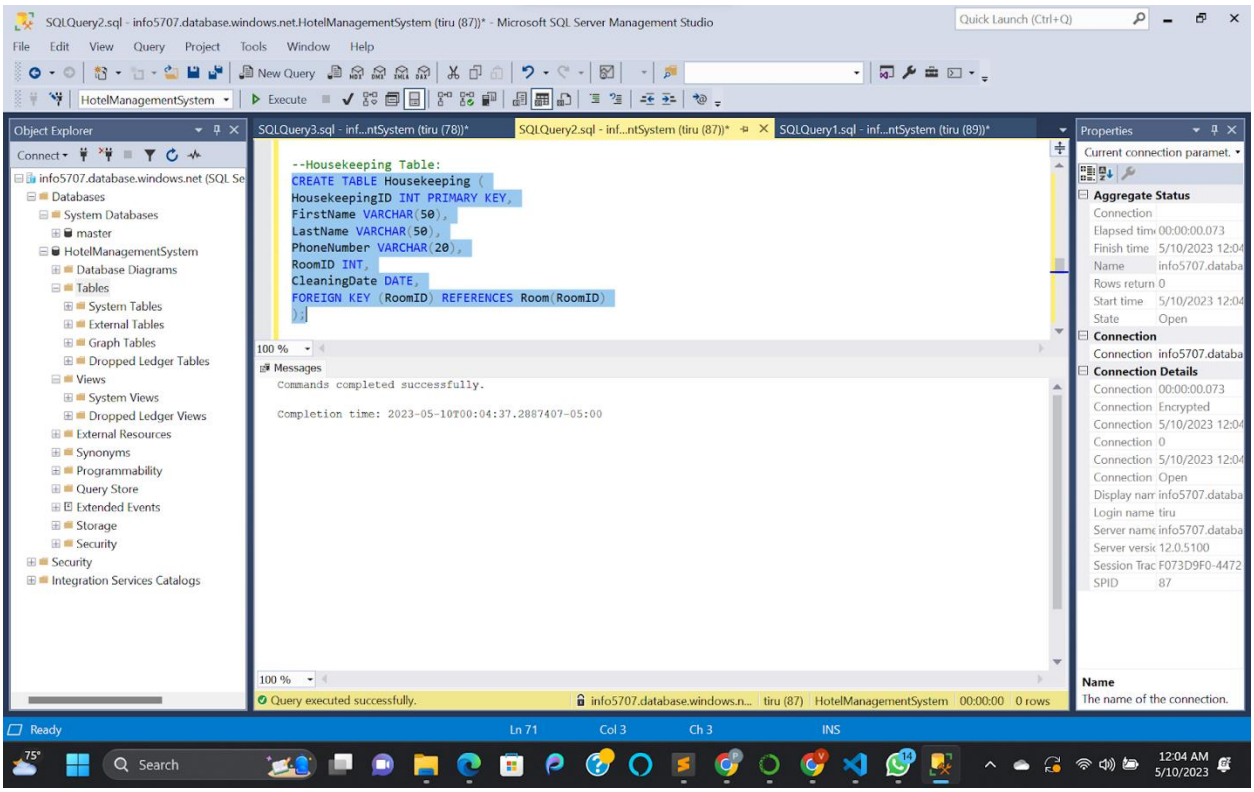

Figure 21: Housekeeping Table Creation

2. Insert Scripts

```
INSERT INTO Housekeeping (HousekeepingID, FirstName, LastName,
PhoneNumber, RoomID, CleaningDate) VALUES
(1, 'Lisa', 'Jones', '555-1234', 1, '2022-06-02'),
(2, 'Mark', 'Davis', '555-5678', 2, '2022-06-03'),
(3, 'Emily', 'Taylor', '555-9012', 3, '2022-06-04'),
(4, 'Tom', 'Brown', '555-3456', 4, '2022-06-05'),
(5, 'Susan', 'Garcia', '555-7890', 5, '2022-06-06'),
(6, 'David', 'Martinez', '555-2345', 6, '2022-06-07'),
```

(7, 'Karen', 'Allen', '555-6789', 7, '2022-06-08'),
(8, 'Mike', 'Young', '555-0123', 8, '2022-06-09'),
(9, 'Mary', 'Hernandez', '555-4567', 9, '2022-06-10'),
(10, 'Adam', 'Lee', '555-8901', 10, '2022-06-11'),
(11, 'Julie', 'Jones', '555-2345', 11, '2022-06-12'),
(12, 'Kevin', 'Davis', '555-6789', 12, '2022-06-13'),
(13, 'Lisa', 'Taylor', '555-0123', 13, '2022-06-14'),
(14, 'Mark', 'Brown', '555-4567', 14, '2022-06-15'),
(15, 'Emily', 'Garcia', '555-8901', 15, '2022-06-16');

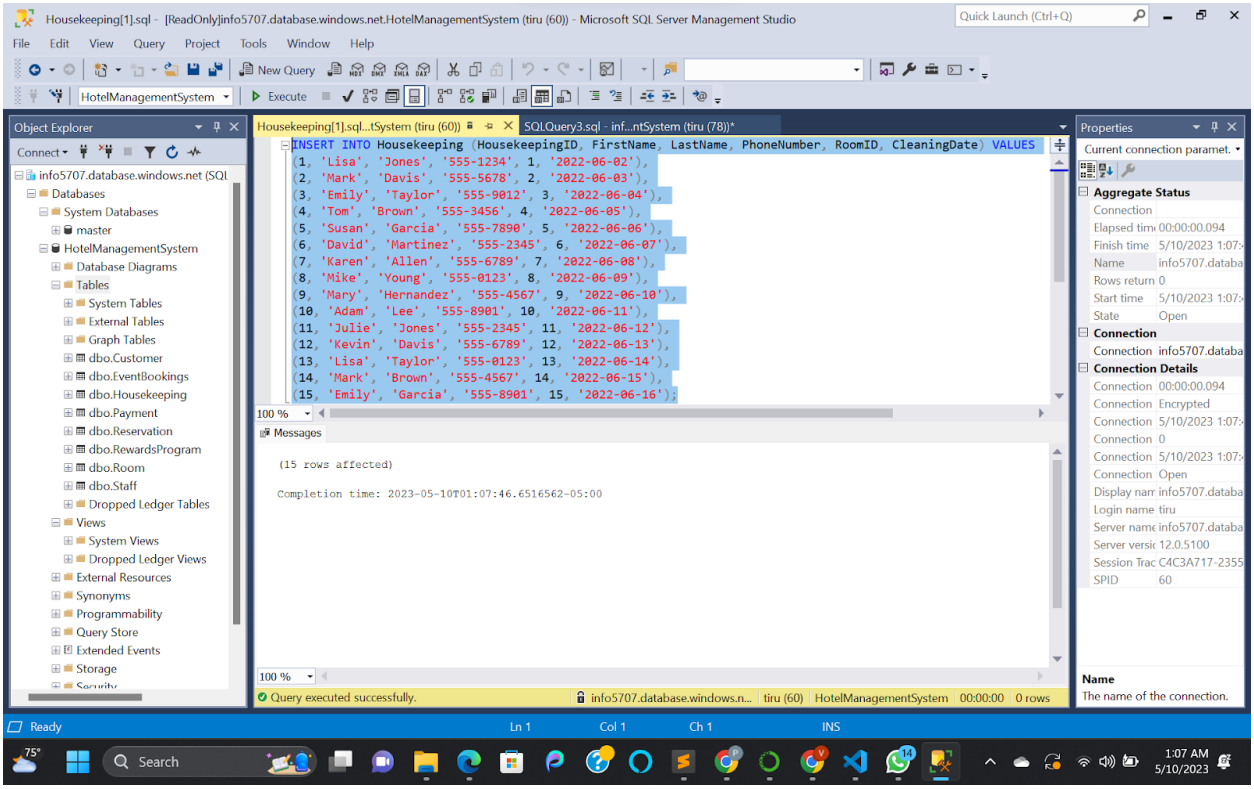

Figure 22: Housekeeping Table Insertions

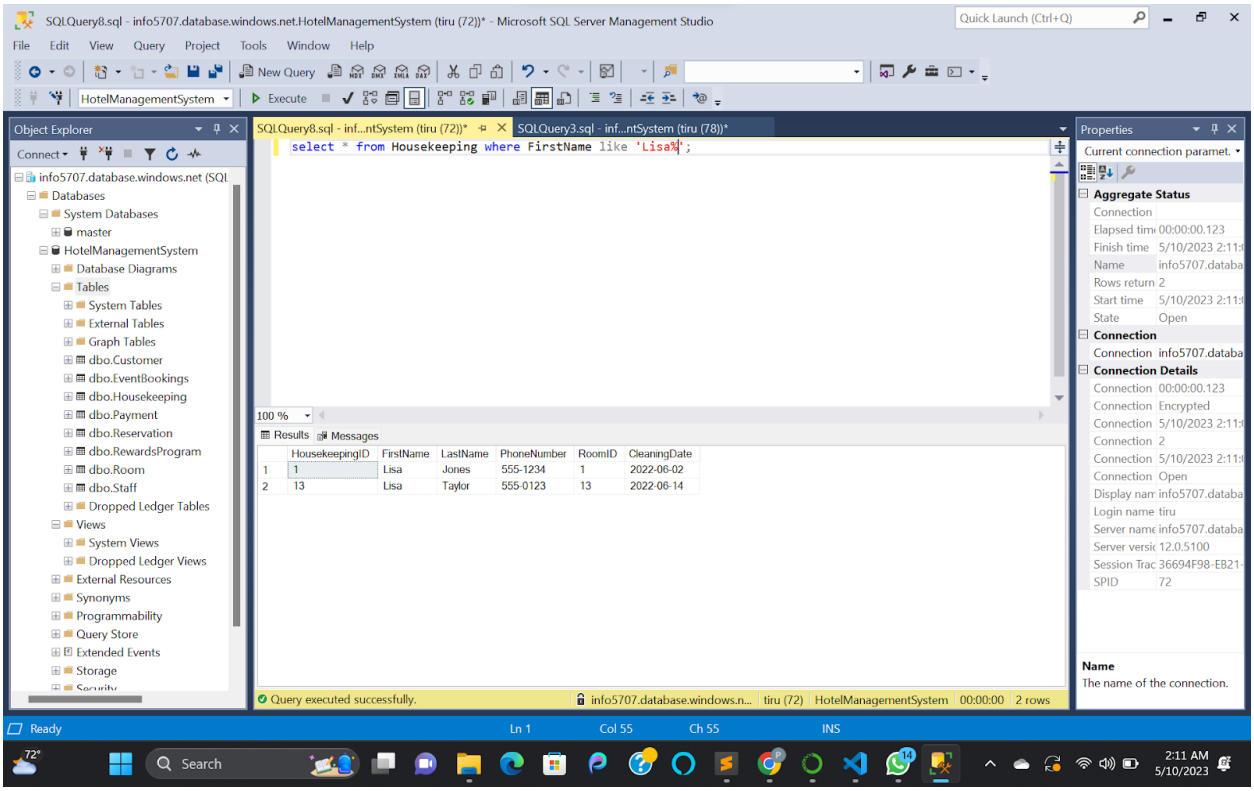

Figure 23: Select Query on Housekeeping Table with Like Operator

g. Rewards Program Table

1. Create Table Statement

```
CREATE TABLE RewardsProgram (
    RewardsID INT PRIMARY KEY,
    CustomerID INT,
    RewardsPoints INT,
    RewardsLevel VARCHAR(50),
    RewardsExpirationDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);
```

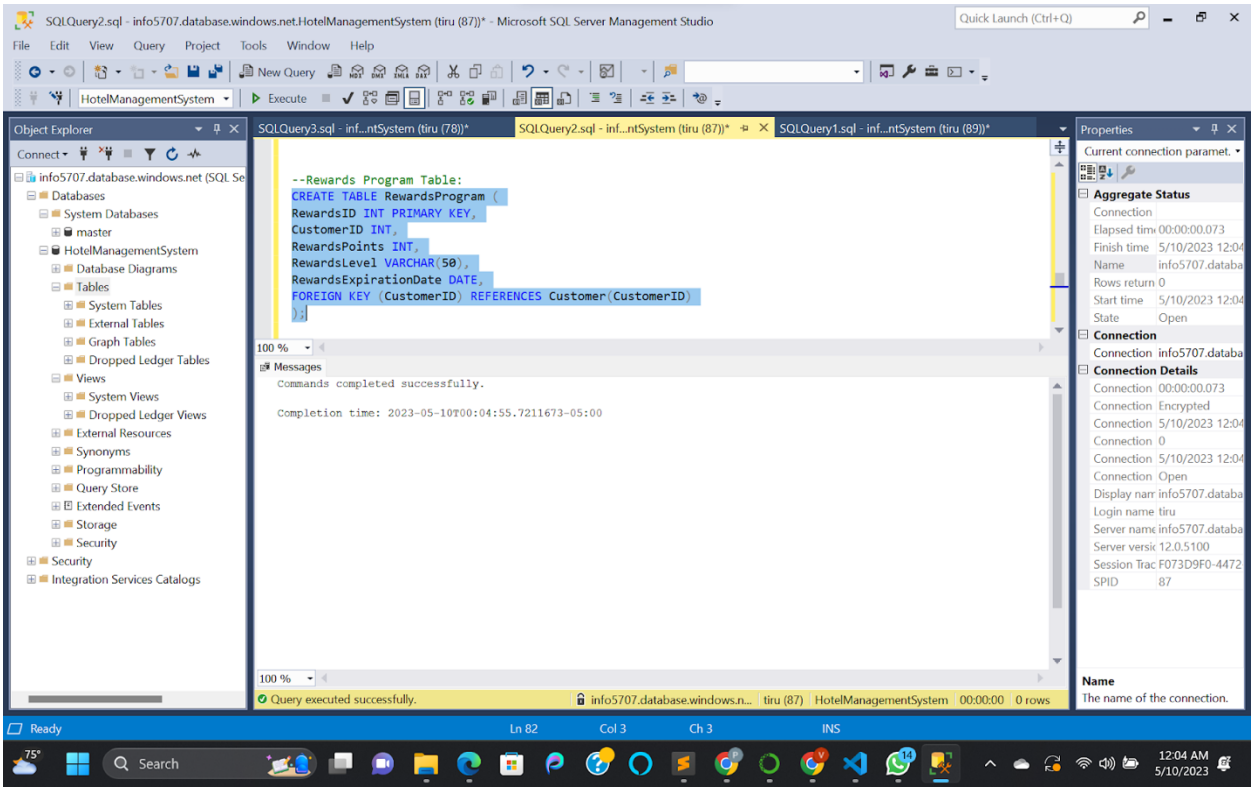

Figure 24: RewardsProgram Table Creation

2. Insert Scripts

```
INSERT INTO RewardsProgram (RewardsID, CustomerID, RewardsPoints,
RewardsLevel, RewardsExpirationDate) VALUES
(1, 1, 100, 'Bronze', '2023-05-31'),
(2, 2, 200, 'Silver', '2023-06-30'),
(3, 3, 150, 'Bronze', '2023-07-31'),
(4, 4, 300, 'Gold', '2023-08-31'),
(5, 5, 200, 'Silver', '2023-09-30'),
(6, 6, 250, 'Gold', '2023-10-31'),
(7, 7, 50, 'None', NULL),
(8, 8, 150, 'Bronze', '2023-11-30'),
(9, 9, 100, 'Bronze', '2023-12-31'),
(10, 10, 200, 'Silver', '2024-01-31'),
(11, 11, 250, 'Gold', '2024-02-28'),
(12, 12, 100, 'Bronze', '2024-03-31'),
(13, 13, 300, 'Gold', '2024-04-30'),
(14, 14, 150, 'Bronze', '2024-05-31'),
(15, 15, 200, 'Silver', '2024-06-30');
```

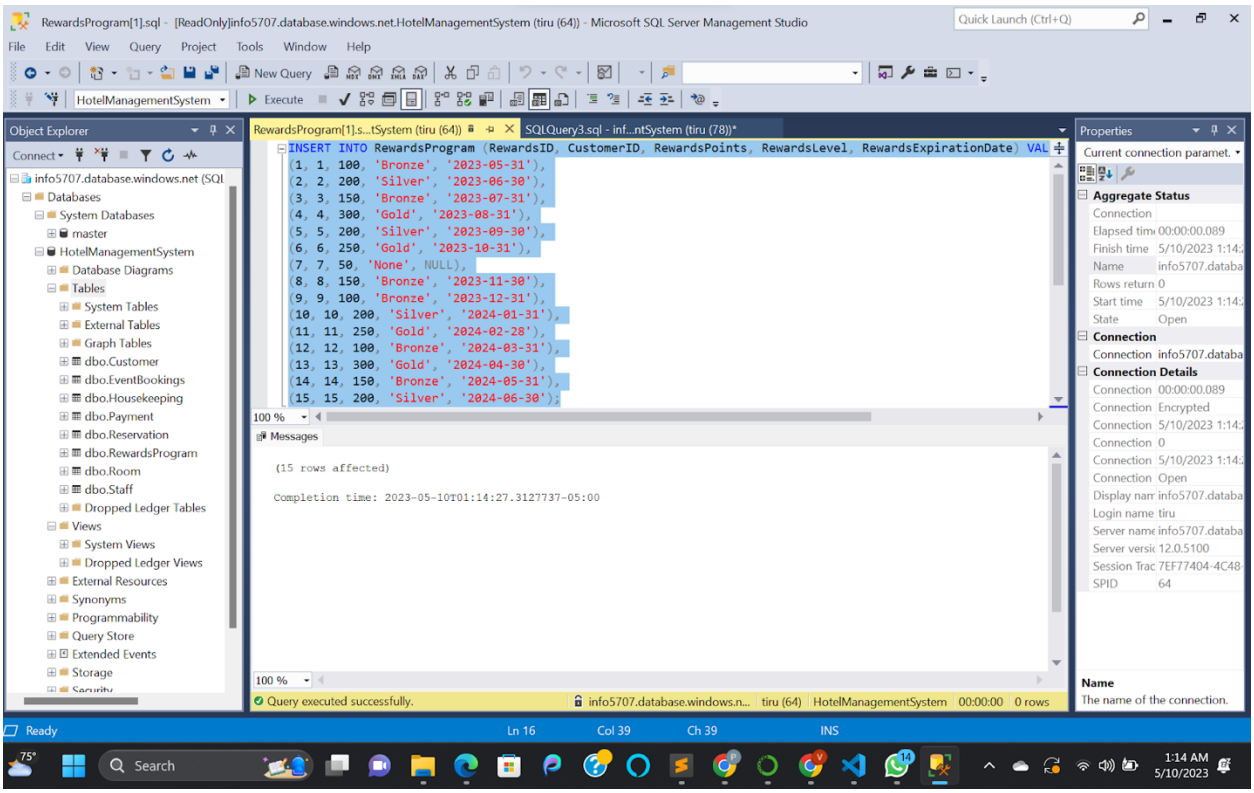

Figure 25: Rewards Program Table Insertions

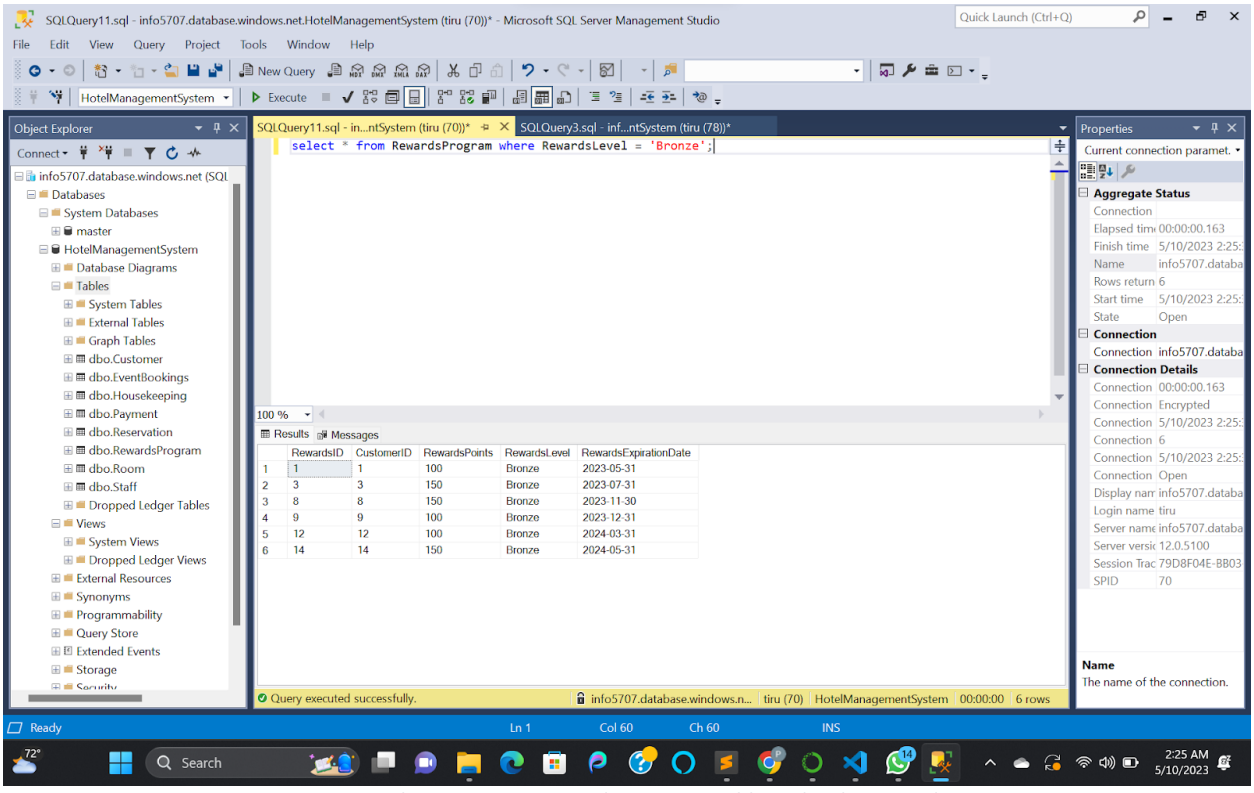

Figure 26: Select Query on Rewards Program Table with Where Condition

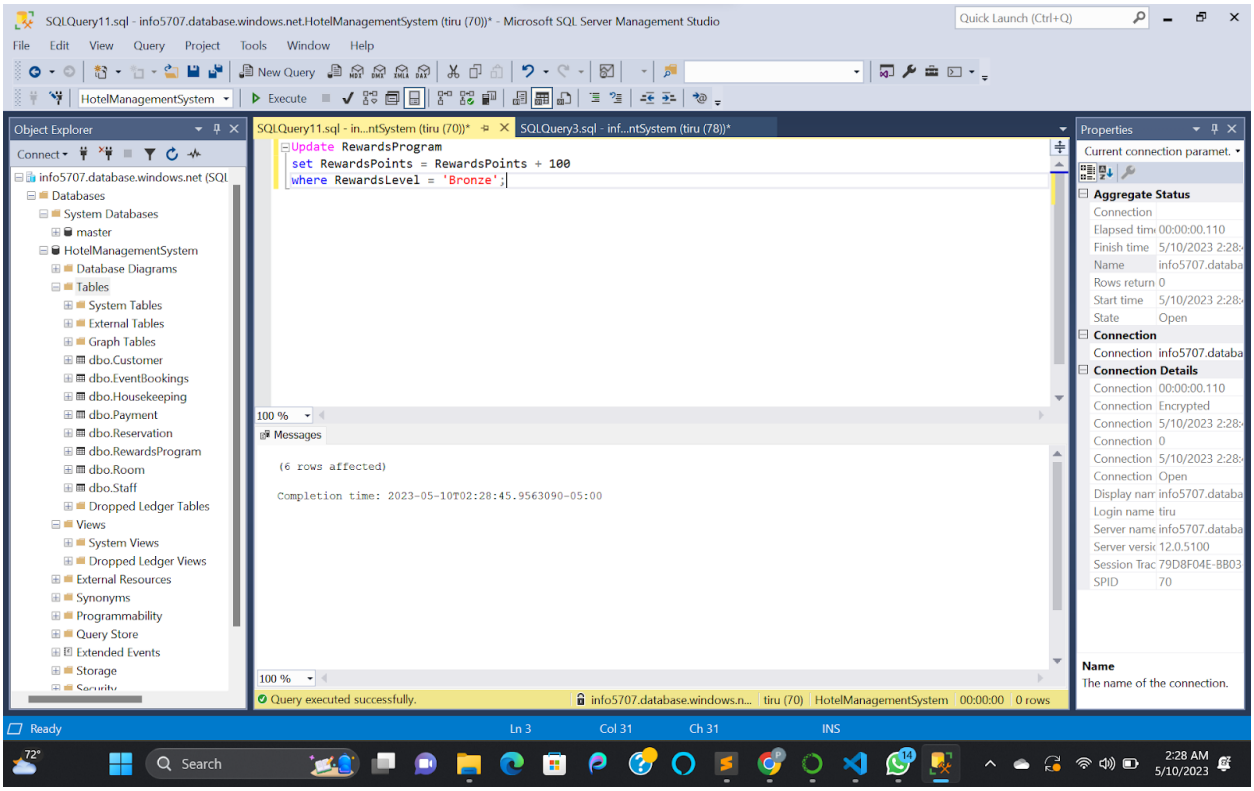

Figure 27: Updating the RewardPoints for a particular Reward Level

h. Event Bookings Table

1. Create Table Statement

```
CREATE TABLE EventBookings (
    EventBookingID INT PRIMARY KEY,
    CustomerID INT,
    EventType VARCHAR(50),
    EventDate DATE,
    NumberOfAttendees INT,
    RoomID INT,
    TotalCost DECIMAL(10,2),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (RoomID) REFERENCES Room(RoomID)
);
```

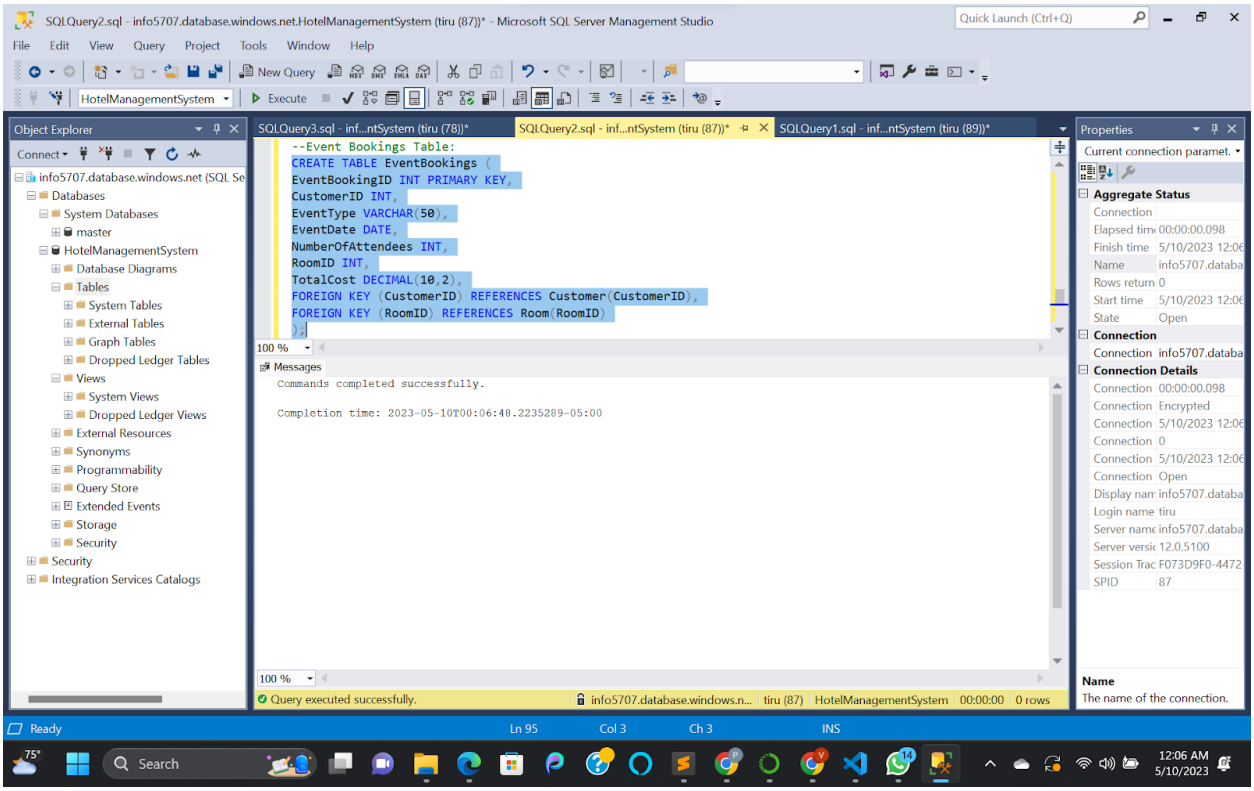

Figure 28: EventBookings Table Creation

2. Insert Scripts

```
INSERT INTO EventBookings (EventBookingID, CustomerID, EventType,
EventDate, NumberOfAttendees, RoomID, TotalCost) VALUES
(1, 1, 'Wedding', '2022-07-15', 100, 6, 4000.00),
(2, 2, 'Business Meeting', '2022-08-01', 50, 3, 1500.00),
(3, 3, 'Conference', '2022-09-10', 200, 5, 8000.00),
(4, 4, 'Birthday Party', '2022-10-15', 20, 2, 500.00),
(5, 5, 'Baby Shower', '2022-11-20', 30, 4, 1000.00),
(6, 6, 'Wedding Anniversary', '2022-12-05', 50, 6, 2000.00),
(7, 7, 'Graduation Party', '2023-01-15', 75, 1, 3000.00),
(8, 8, 'Business Meeting', '2023-02-20', 30, 3, 1000.00),
(9, 9, 'Conference', '2023-03-30', 150, 5, 6000.00),
(10, 10, 'Birthday Party', '2023-04-05', 10, 2, 250.00),
(11, 11, 'Baby Shower', '2023-05-10', 20, 4, 500.00),
(12, 12, 'Wedding Anniversary', '2023-06-15', 50, 6, 2000.00),
(13, 13, 'Graduation Party', '2023-07-20', 100, 1, 4000.00),
(14, 14, 'Business Meeting', '2023-08-25', 40, 3, 1500.00),
(15, 15, 'Conference', '2023-09-30', 250, 5, 10000.00);
```

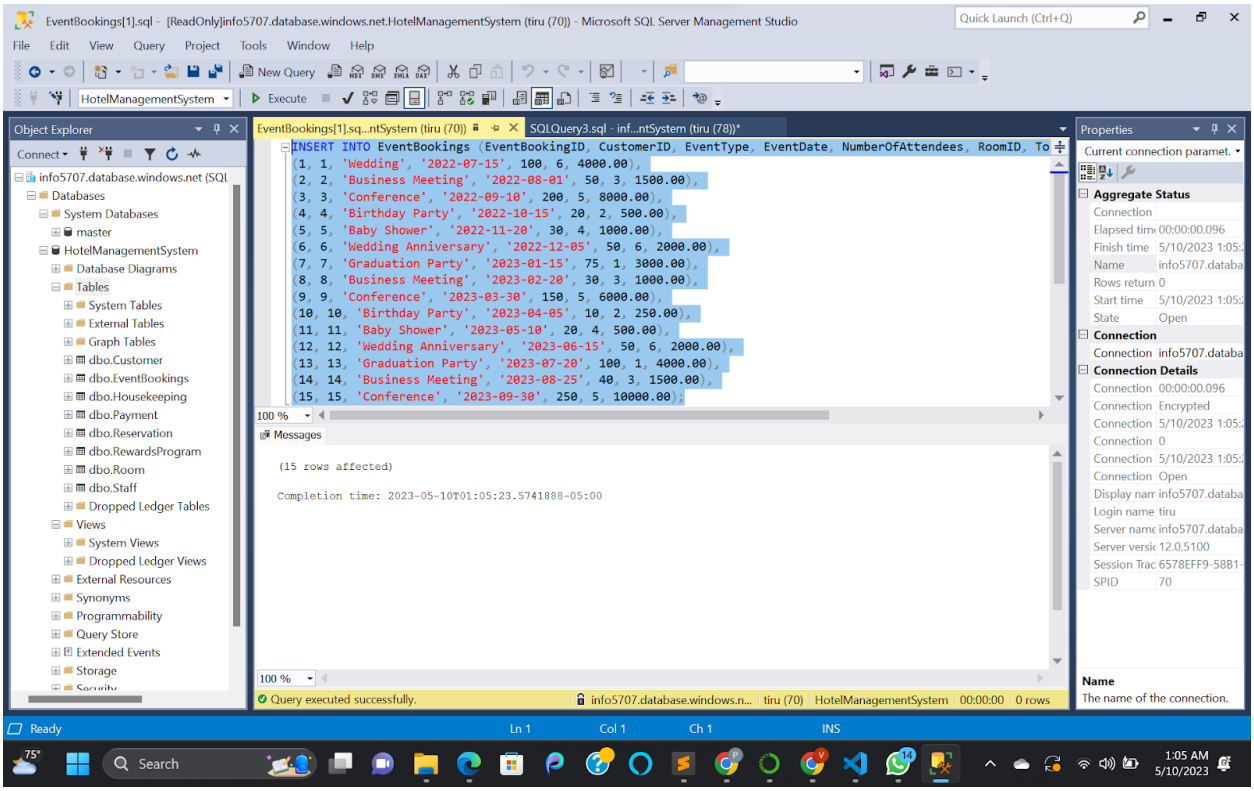

Figure 29: Event Bookings Insertions

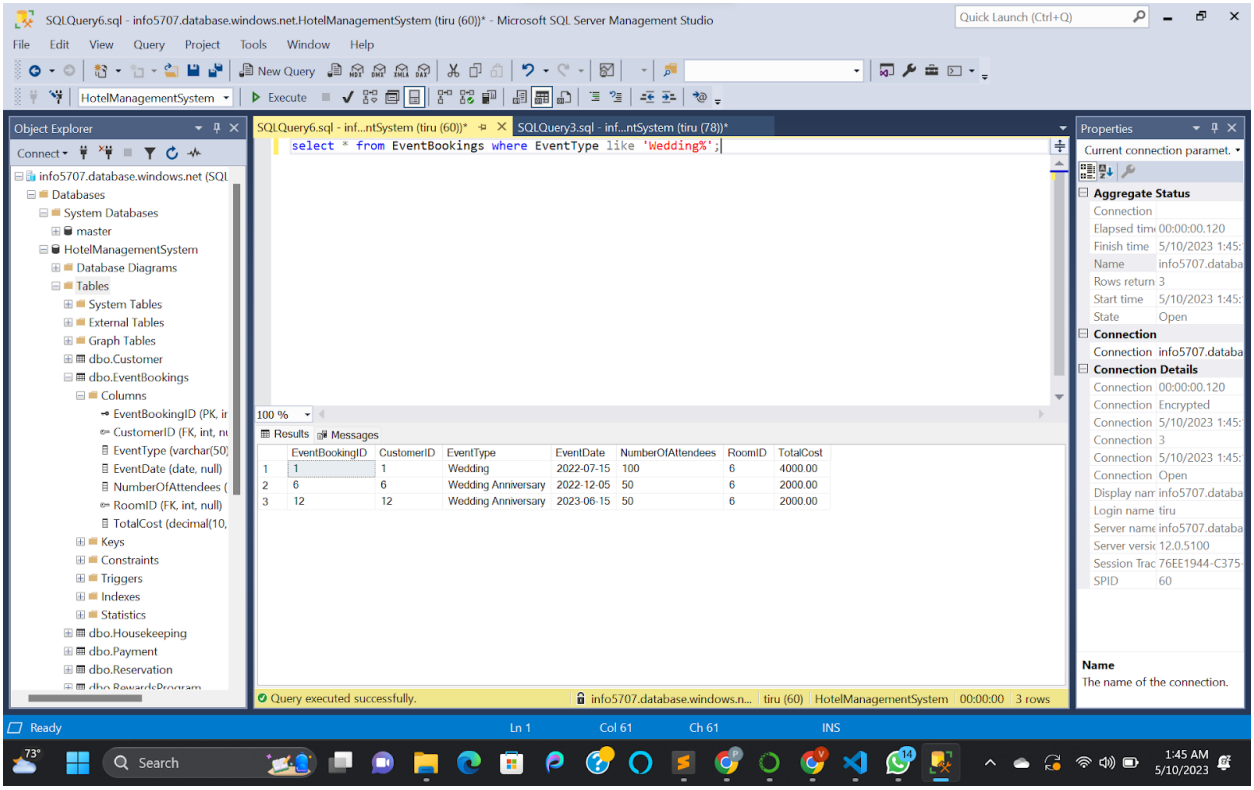

Figure 30: Select with Like Operator for EventBookings Table

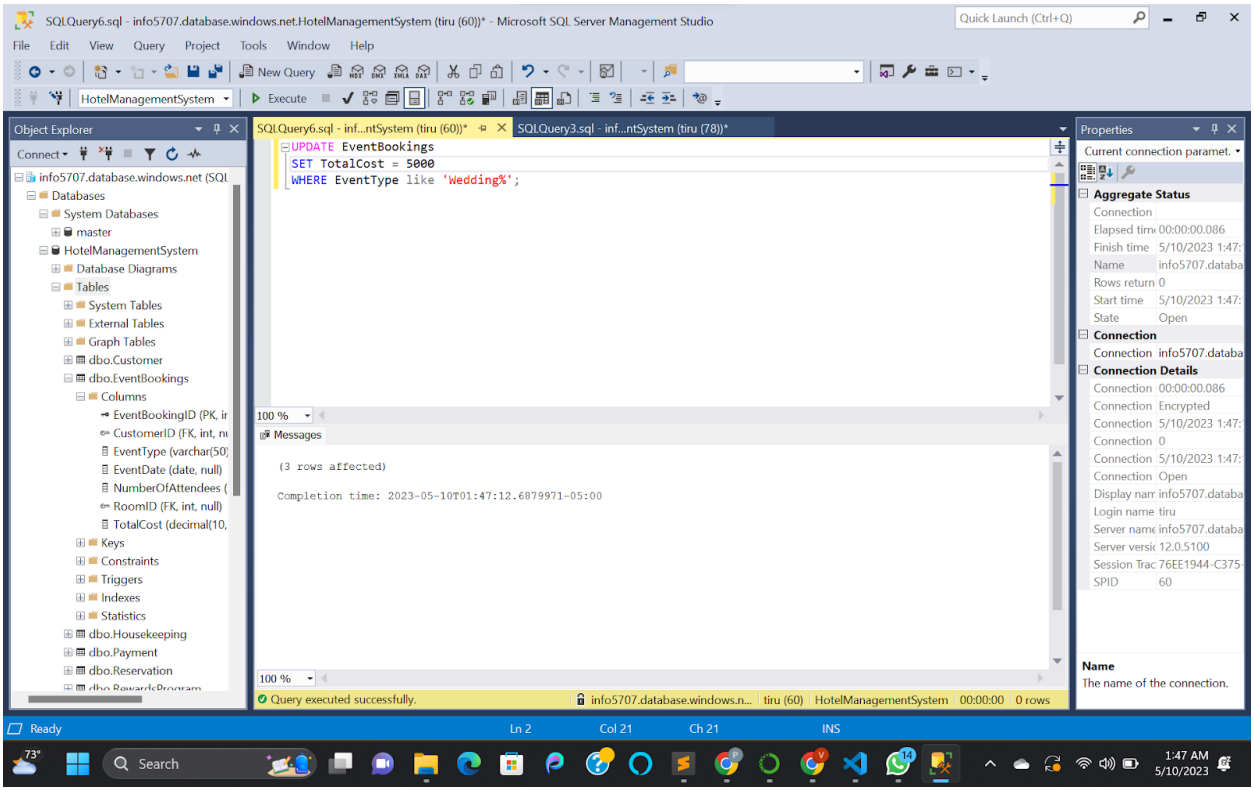

Figure 31: Updating Cost on EventBookings Table

Operational Queries:

1. Retrieve the details of all reservations along with the names of the customers who made them.

```
SELECT
        Reservation.*,
        Customer.FirstName,
        Customer.LastName
FROM Reservation
JOIN Customer ON Reservation.CustomerID = Customer.CustomerID;
```

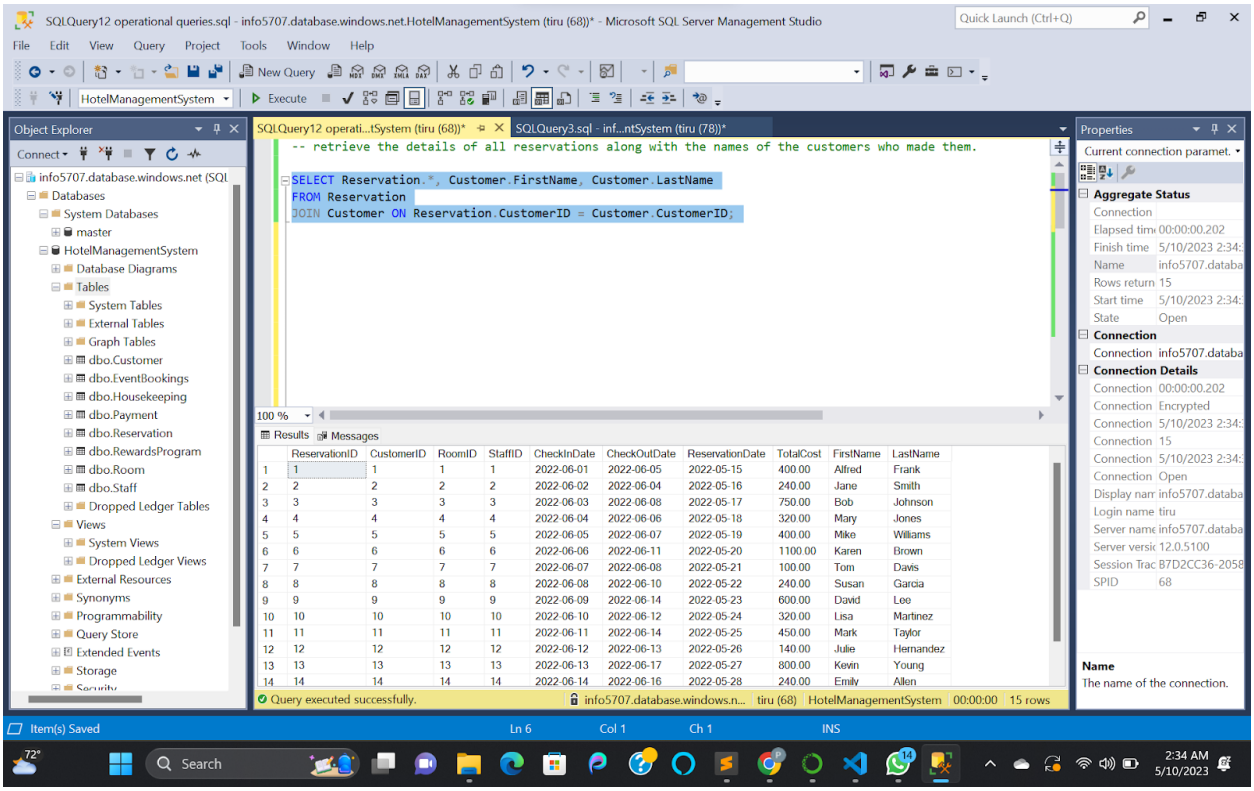

Figure 32: Details of all reservations along with the names of the customers who made them

2. Retrieving the details of all event bookings along with the room number and type of the room where the event will be held.

```
SELECT
        EventBookings.*,
        Room.RoomNumber,
        Room.RoomType
FROM EventBookings
JOIN Room ON EventBookings.RoomID = Room.RoomID;
```

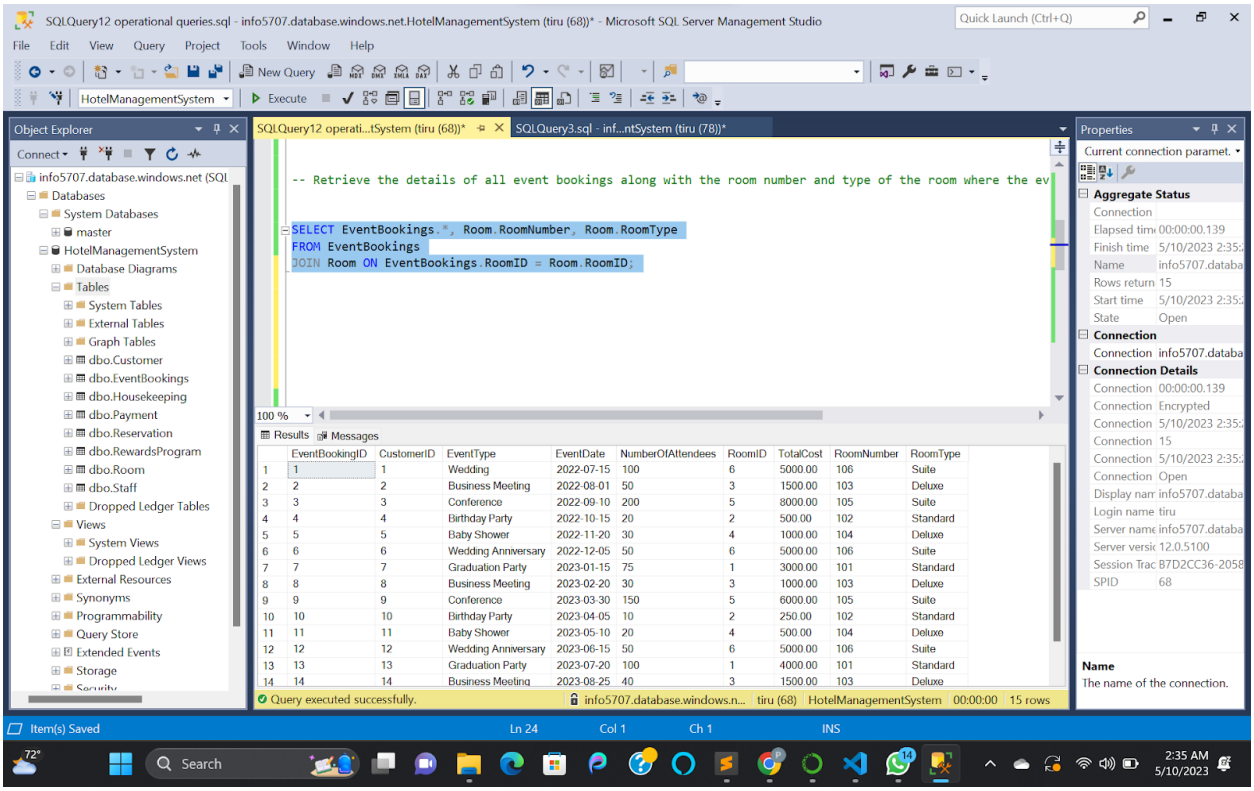

Figure 33: Details of all event bookings along with the room number and type of the room where the event will be held.

3. Retrieve the details of all payments along with the names of the customers who made the reservations.

```
SELECT
        Payment.*,
        Customer.FirstName,
        Customer.LastName
FROM Payment
JOIN Reservation ON Payment.ReservationID = Reservation.ReservationID
JOIN Customer ON Reservation.CustomerID = Customer.CustomerID;
```

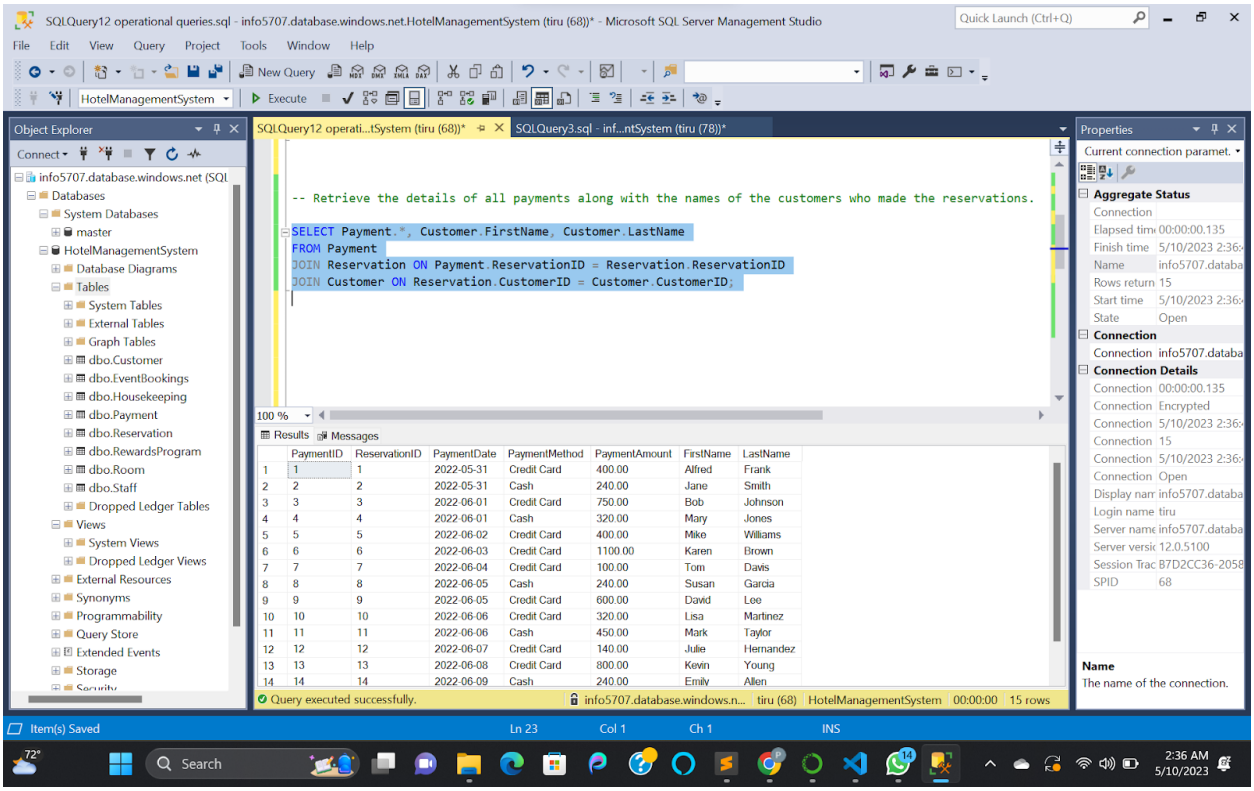

Figure 34: details of all payments along with the names of the customers who made the reservations.

4. Retrieve the details of all reservations that were made by customers who have a rewards level of “Gold”

```
SELECT Reservation.*
FROM Reservation
JOIN Customer ON Reservation.CustomerID = Customer.CustomerID
JOIN RewardsProgram ON Customer.CustomerID = RewardsProgram.CustomerID
WHERE RewardsProgram.RewardsLevel = 'Gold';
```

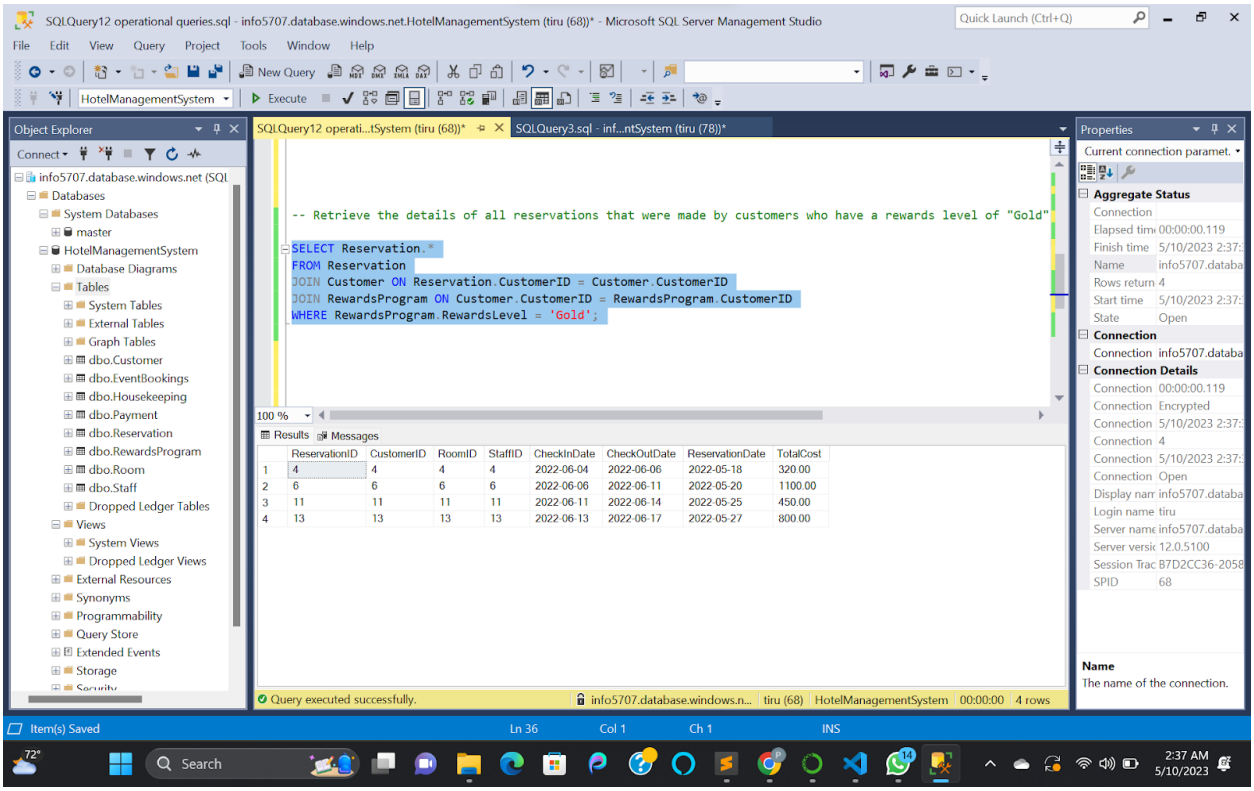

Figure 35: names of all staff members who have ever cleaned a room that is currently occupied by a customer.

5. Retrive the names of all staff members who have ever cleaned a room that is currently occupied by a customer.

```
SELECT
        DISTINCT Staff.FirstName, Staff.LastName
FROM Staff
JOIN Housekeeping ON Staff.StaffID = Housekeeping.HousekeepingID
JOIN Room ON Housekeeping.RoomID = Room.RoomID
JOIN Reservation ON Room.RoomID = Reservation.RoomID
WHERE Reservation.CheckOutDate > GETDATE();
```

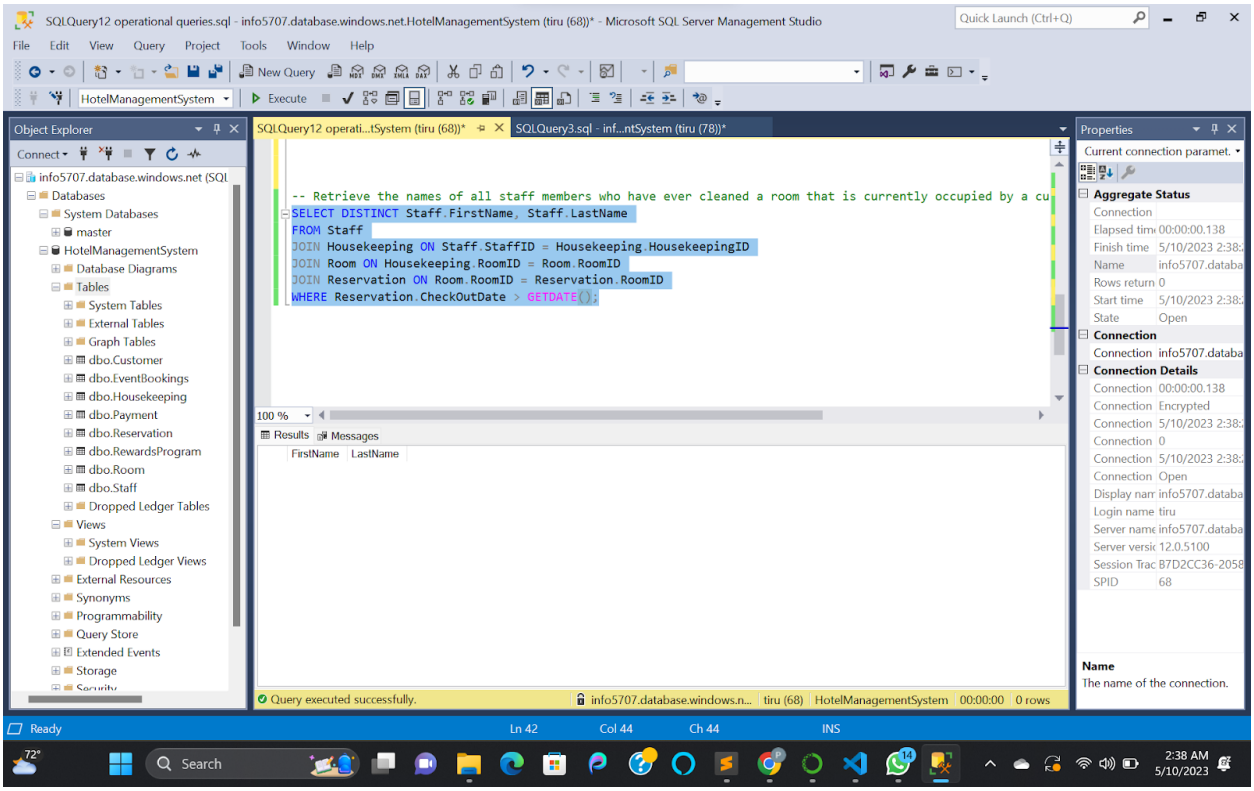

Figure 36: names of all staff members who have ever cleaned a room that is currently occupied by a customer.